



**Hi3516CV610 UHD Smart Vision SoC**

## **Data Sheet**

**Issue** 00B03

**Date** 2024-09-15

**Copyright © HiSilicon (Shanghai) Technologies Co., Ltd. 2024. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of HiSilicon (Shanghai) Technologies Co., Ltd.

**Trademarks and Permissions**



, **HISILICON**, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

**Notice**

The purchased products, services and features are stipulated by the contract made between HiSilicon 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. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

**HiSilicon (Shanghai) Technologies Co., Ltd.**

Address: Room 101, 2 Hongqiaogang Road,  
Qingpu District,  
Shanghai 201721 P. R. China

Website: <http://www.hisilicon.com/en/>

Email: [support@hisilicon.com](mailto:support@hisilicon.com)



## Contents

---

<b>About This Document .....</b>	<b>1</b>
----------------------------------	----------



# About This Document

## Purpose

This document describes the features, logical structures, functions, operating modes, and related registers of each module of Hi3516CV610. This document also describes the interface timings and related parameters in diagrams. In addition, this document describes the pins, pin usages, performance parameters, and package dimension of Hi3516CV610 in detail.

### NOTE

Hi35xxVxxx in this document refers to Hi3516CV610. Hi3516CV610 is classified into the 00B, 10B, 20B, 00S, 20S, 00G, and 20G models based on functions. For details about the differences, see section "Model Configuration Differences."

## Related Version

The following table lists the product version related to this document.

Product Name	Version
Hi3516C	V610

## Intended Audience

This document is intended for:

- Design and maintenance personnel for electronics
- Sales personnel for electronic parts and components



# Conventions

## Table Content Conventions

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

Content	Description
-	The cell is blank.
*	The content in this cell is configurable.

# Notes

## Register Attributes

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

Symbol	Description	Symbol	Description
RO	The register is read-only.	RW	The register is read/write.
RC	The register is cleared on a read.	WC	The register can be read. The register is cleared when 1 is written. The register keeps unchanged when 0 is written.
RW_CLR	The register is readable and writable. It is cleared by the logic.	RW_LOCK	The register is readable and writable. It is locked after 1 is written.
RW_SEC	The secure CPU can read and write the register.	RW_SEC_LOCK	The secure CPU can read and write the register. The register is locked after 1 is written.

# Numerical System

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



Type	Symbol	Value
Data capacity (such as the RAM capacity)	K	1024
	M	1,048,576
	G	1,073,741,824
Frequency, data rate	k	1000
	M	1,000,000
	G	1,000,000,000

The expressions of addresses and data are described as follows.

Symbol	Example	Description
0x	0xFE04, 0x18	Address or data in hexadecimal
0b	0b000, 0b00 00000000	Data or sequence in binary (register description is excluded.)
X	00X, 1XX	In data expression, X indicates 0 or 1. For example, 00X indicates 000 or 001, and 1XX indicates 100, 101, 110, or 111.

## Change History

Changes between document issues are cumulative. The latest document issue contains all changes made in earlier issues.

### Issue 00B03 (2024-09-15)

This issue is the third draft release, which incorporates the following changes:

#### Chapter 2 Hardware

Section 2.5.1 and 2.5.3 are modified.

#### Chapter 3 System

Section 3.2.3.1, 3.2.4, 3.2.5, and 3.10.5 are modified.

Section 3.5.3.2 is deleted.



In section 4.8, the registers SC\_LOCKEN, WDG\_RST\_CNT, SOFTRST\_CNT, and RESET\_FLAG are deleted.

### **Chapter 12 Peripherals**

In section 12.3.5, added NOTE to the AT\_STAT\_R register.

Section 12.8 and 12.9 are modified.

## **Issue 00B02 (2024-06-20)**

This issue is the second draft release, which incorporates the following changes:  
“Ordering Information” is added.

### **Chapter 3 System**

Section 3.5.4 is modified.

## **Issue 00B01 (2024-05-10)**

This issue is the first draft release.



# Contents

<b>1 Product Description .....</b>	<b>1-1</b>
1.1 Introduction .....	1-1
1.2 Architecture .....	1-1
1.2.1 Functional Block Diagram.....	1-1
1.2.2 Processor Core.....	1-2
1.2.3 Intelligent Video Engine (IVE) .....	1-2
1.2.4 AI ISP.....	1-3
1.2.5 Video Processing.....	1-3
1.2.6 Video Encoding .....	1-3
1.2.7 VI Interfaces .....	1-4
1.2.8 Audio Interfaces and Processing.....	1-4
1.2.9 Security Isolation and Engine .....	1-4
1.2.10 Peripheral Interfaces.....	1-5
1.2.11 External Memory Interfaces.....	1-5
1.2.12 SDK.....	1-5
1.2.13 Physical Specifications.....	1-5
1.2.14 Model Configuration Differences.....	1-6
1.3 Boot and Upgrade Modes .....	1-7
1.3.1 Overview.....	1-7
1.3.2 Relationship Between Boot Media and Power-On Latch Values.....	1-7
1.3.3 Boot Modes .....	1-9
1.4 Address Space Mapping.....	1-9



## Figures

---

**Figure 1-1 Functional block diagram of Hi3516CV610 .....1-2**



## Tables

---

<b>Table 1-1 Boot media</b> .....	1-8
<b>Table 1-2 Address space mapping</b> .....	1-9



# 1

## **Product Description**

### **1.1 Introduction**

Hi3516CV610 is an IPC SoC designed for the surveillance market. It is mainly used for indoor and outdoor box cameras, dome cameras, fixed dome cameras, turret cameras, box-and-dome cameras, and dual-lens varifocal cameras. It powers solutions and products with competitive edges, leading the industry development in such fields as open OS, next-generation video codec standards, cyber security, privacy protection, and AI.

#### **DNOTE**

Hi3516CV610 is classified into the 00B, 10B, 20B, 00S, 20S, 00G, and 20G models based on functions. This document uses Hi3516CV610-20S as an example. For details about the differences, see section "[Model Configuration Differences](#)."

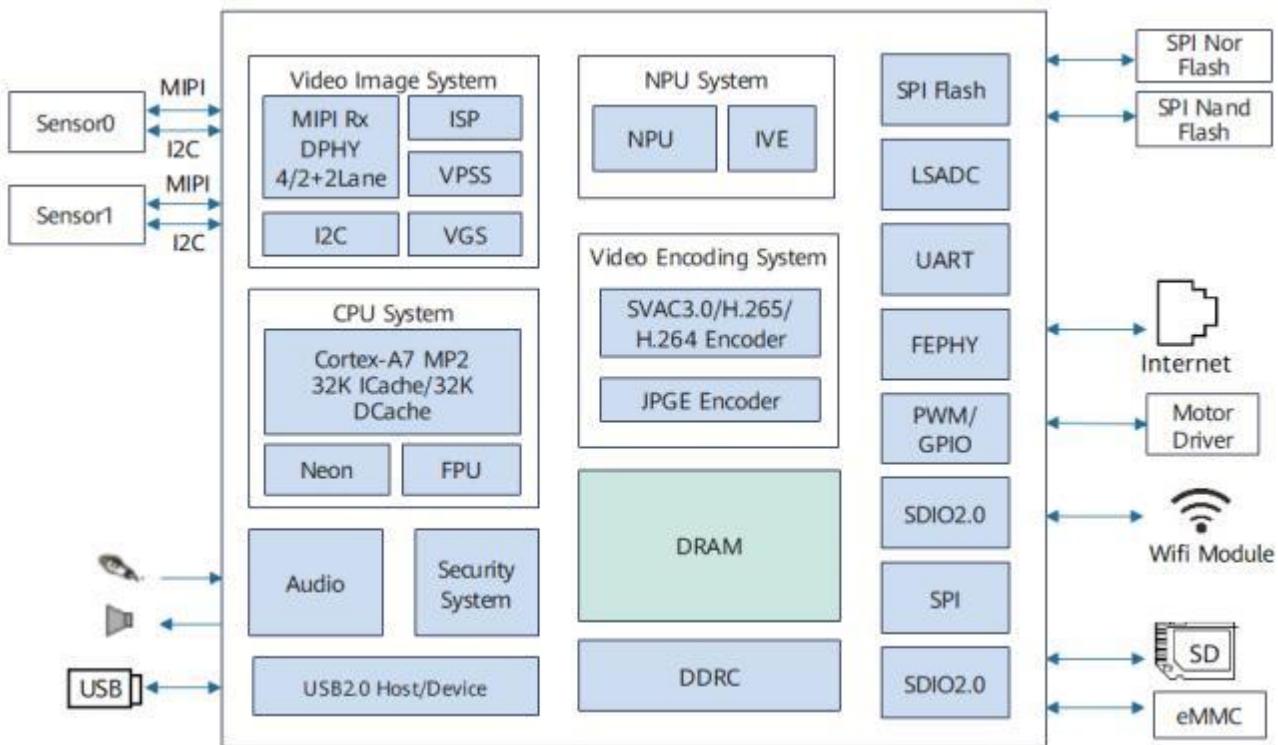
### **1.2 Architecture**

#### **1.2.1 Functional Block Diagram**

[Figure 1-1](#) shows the functional block diagram of Hi3516CV610.



Figure 1-1 Functional block diagram of Hi3516CV610



## 1.2.2 Processor Core

- Arm Cortex-A7 MP2
- Clock rate: 950 MHz
- 32 KB I-cache, 32 KB D-cache, and 128 KB L2 cache
- Neon and FPU

## 1.2.3 Intelligent Video Engine (IVE)

- Neural network processing unit (NPU)
  - 1 TOPS compute performance
  - Applicability to hundreds of mainstream algorithms in the industry, assisting customers in rapid commercial use
  - Transformer feature acceleration and built-in dedicated multimodal large model
  - Efficient model production and evolution
  - Built-in detection algorithms for human faces, human figures, vehicles, parcels, and pets
- IVE2.5 upgraded operators, supporting motion detection, perimeter protection, tracking, perspective transformation, video diagnosis, and multiple intelligent analysis applications



## 1.2.4 AI ISP

- Basic ISP functions
  - Dynamic defect pixel correction
  - Adjustable 3A functions (AE, AWB, and AF)
  - Lens shading correction
- High dynamic range
  - 2-frame WDR combination
  - Advanced local tone mapping
  - Highlight compensation and backlight compensation
- Clarity and noise reduction (NR)
  - Fixed-pattern noise (FPN) reduction
  - Effect enhancement algorithm for motion regions
  - Temporal and spatial NR
  - Multi-level motion detection
  - Region-based enhancement
  - Chrominance NR 2.0
- Color management and contrast enhancement
  - Purple fringing removal 2.0
  - Key color enhancement
  - Local dynamic contrast improvement (DCI)
  - Dehaze
- PQTools 1.2

## 1.2.5 Video Processing

- Image scaling by 1/15.5x to 16x
- Lens distortion correction (LDC)
- Image mirroring, flipping, and rotation by 90 or 270 degrees

## 1.2.6 Video Encoding

- SVAC 3.0 Main Profile Level: 8.0.30
- H.265 Main Profile Level 5
- H.264 BP/MP/HP Level 5.1
- Maximum encoding resolution: 3840 x 2160
- I-frames and P-frames
- Intelligent encoding 2.0
- Privacy-oriented encoding



- The typical performance of multi-stream encoding is as follows:
  - 3200 x 1800@30 fps+1280 x 720@30 fps
  - 3840 x 2160@20 fps+1280 x 720@20 fps
- CBR, VBR, ABR, AVBR, CVBR, QVBR, FixQp, and QpMap bit rate control modes
- Maximum output bit rate: 80 Mbit/s
- OSD overlaying before encoding of eight regions
- Encoding of eight ROIs
- JPEG Baseline encoding
- Maximum resolution for JPEG encoding: 4096 x 4096
- Maximum performance for JPEG encoding:
  - 3200 x 1800@30 fps (YUV420)
  - 3840 x 2160@20 fps (YUV420)

## 1.2.7 VI Interfaces

- MIPI/LVDS/Sub-LVDS/HiSPi interfaces
- Combinations such as 2x2-lane or 4-lane modes, and up to 2-sensor inputs
- 8-/10-/12-bit RGB Bayer
- 8-/10-/12-bit DVP
- BT.601, BT.656 and BT.1120 VI interfaces

## 1.2.8 Audio Interfaces and Processing

- Built-in audio codec, supporting 16-bit dual-channel differential voice inputs and single-channel single-ended voice outputs
- One I<sup>S</sup>S interface, compatible with the multi-channel time division multiplexing (TDM) transmission mode
- Audio codec complying with multiple protocols
- 3A (AEC/ANR/ALC) processing 2.0
- Wind noise reduction algorithm
- Baby crying detection 2.0, with improved wake-up rate

## 1.2.9 Security Isolation and Engine

- Secure boot
- REE and TEE hardware isolation solution based on TrustZone
- Multiple encryption and decryption algorithms implemented by hardware, such as AES, RSA, ECC, and SHA
- Hardware-based SM2/3/4 Chinese cryptographic algorithms



## 1.2.10 Peripheral Interfaces

- Integrated FEPHY, supporting TSO network acceleration
- Two SDIO 2.0 interfaces
  - SDXC card
  - Connection with the Wi-Fi module
- One USB 2.0 host or device interface
- Power-on reset (POR) and external input reset
- Integrated RTC with independent power supply
- Integrated multi-channel LSADCs
- Multiple UART, I<sup>2</sup>C, SPI, PWM, and GPIO interfaces

## 1.2.11 External Memory Interfaces

- SDRAM interface  
Built-in 1 Gbit DDR3/3L rate: up to 2133 Mbit/s
- SPI NOR flash interface
  - 1-/2-/4-wire mode
  - Maximum capacity: 32 MB
- SPI NAND flash interface
  - 1-/2-/4-wire mode
  - Maximum capacity: 512 MB
- eMMC 4.5 interface, with 4-bit data width
- Boot from the eMMC or SPI NOR/SPI NAND flash

## 1.2.12 SDK

- Linux 5.10 SDK
- OpenHarmony 4.1

## 1.2.13 Physical Specifications

- Power consumption  
Typical power consumption: 1043 mw
- Operating voltages
  - Typical core voltage: 0.9 V
  - I/O voltage: 1.8 V or 3.3 V
  - DDR3 and DDR3L interface voltages: 1.5 V and 1.35 V, respectively
- Operating temperature: -20°C to +70°C
- Package:



- QFN package (9 mm x 9 mm), RoHS compliant
- Pin pitch: 0.35 mm

### 1.2.14 Model Configuration Differences

Category	Hi3516CV610											
	10B	20B	20S	20G	00B	00S	00G					
Maximum performance	5M@30 fps	4K@20 fps or 6M@30 fps										
Maximum resolution	2880 x 1620	3840 x 2160										
Compute power	0.5 Tops	1 Tops										
Encoding protocol	SVAC3.0/ H.265/H.264 <b>64 (SVAC 3.0 does not support SmartCRR)</b>	H.265/ H.264	SVAC 3.0/H.265/H.264		H.265/ H.264	SVAC 3.0/ H.265/H.264						
Maximum encoding capability	5M+D1	6M+720P										
WDR	Not supported	Supported										
DDR	Built-in DDR2: 1333 Mbit/s	Built-in DDR3/3L: 2133 Mbit/s			External DDR3/3L: 2133 Mbit/s							
	512 Mbit	1 Gbit			Up to 4 Gbit							
eMMC	eMMC 4.5 (4-wire)			eMMC 5.0 (8-wire)								
Package	QFN9 x 9 (0.35 mm)			TFBGA12 x 13.3 (0.65 mm)								
Security specifications	-		Support GB35114	-		Support GB35114						



## 1.3 Boot and Upgrade Modes

### 1.3.1 Overview

Hi3516CV610 has an embedded boot ROM (BootROM), which starts the boot program after the chip reset is cleared.

#### Boot Medium

Hi3516CV610 contains multiple peripheral interfaces, which can be used as boot media interfaces:

- SPI NAND or SPI NOR flash storage interface
- eMMC storage interface

The selection of the boot medium is determined by the SFC\_EMMC\_BOOT\_MODE, BOOT\_SEL1, or BOOT\_SEL0 signal value.

#### Burning (Upgrade)

The chip can burn (upgrade) boot media through an SD card, USB, or UART port. The upgrade mode of the SD card and USB is determined by the UPDATE\_MODE signal value, and the UART burning is determined by the FAST\_BOOT\_MODE signal value.

##### NOTE

- When the chip boots from the eMMC, upgrade cannot be implemented by connecting SDIO0 to the SD card.
- When connecting SDIO1 to the SD card for upgrade, burn `sdio0_selfboot_disable` (OTP) to disable the SDIO0 upgrade.

### 1.3.2 Relationship Between Boot Media and Power-On Latch Values

The boot or upgrade mode is determined by FAST\_BOOT\_MODE, SFC\_EMMC\_BOOT\_MODE, BOOT\_SEL1, BOOT\_SEL0, and UPDATE\_MODE\_N signals.



### NOTE

- The FAST\_BOOT\_MODE signal is the power-on latch value of the SENSOR1\_CLK pin.
- The SFC\_EMMC\_BOOT\_MODE signal is the power-on latch value of the SENSOR0\_CLK pin.
- The BOOT\_SEL1 signal is the power-on latch value of the GPIO0\_1 pin.
- The BOOT\_SEL0 signal is the power-on latch value of the SFC\_CLK pin.
- The UPDATE\_MODE\_N signal shows the status of GPIO0\_0 during system startup. Generally, GPIO0\_0 can be designed into a key. Value **0** indicates that the key is pressed and GPIO0\_0 is in upgrade mode. Value **1** indicates that the key is unpressed and GPIO0\_0 is in non-upgrade mode.
- The preceding power-on latch values are not affected by the system soft reset.
- SFC\_EMMC\_BOOT\_MODE, BOOT\_SEL1, and BOOT\_SEL0 determine the target media for booting and upgrading.
- FAST\_BOOT\_MODE specifies whether to enter the burning mod over the serial port.

The status of FAST\_BOOT\_MODE, SFC\_EMMC\_BOOT\_MODE, BOOT\_SEL1, and BOOT\_SEL0 can be obtained by system control registers SYSSTAT[4:2] and SYSSTAT[11].

### NOTE

- For details, see the description of the SYSSTAT registers in chapter "System."
- For details about the signal names, see the *Hi35xxVxxx\_PINOUT\_EN*.

[Table 1-1](#)describes the relationship between boot media and signals.

**Table 1-1** Boot media

UPDATE_MODE	FAST_BOOT_MODE	BOOT_SEL1	BOOT_SEL0	Boot Medium
1	0	0	0	Booting from the SPI NOR flash
1	0	0	1	Booting from the SPI NAND flash
1	0	1	1	Booting from the eMMC
1	1	0	0	Burning an image to the SPI NOR flash over a serial port
1	1	0	1	Burning an image to the SPI NAND flash over a serial port
1	1	1	1	Burning an image to the eMMC over a serial port



UPDATE_MODE	FAST_BOOT_MODE	BOOT_SEL1	BOOT_SEL0	Boot Medium
0	X	0	0	Upgrading the SPI NOR flash from the SDIO or USB
0	X	0	1	Upgrading the SPI NAND flash from the SDIO or USB
0	X	1	1	Upgrading the eMMC flash from the SDIO or USB

### 1.3.3 Boot Modes

The chip supports the secure boot function.

- Secure boot verification is supported to ensure that only verified images can run during system boot.
- The system boot image can be encrypted to ensure that only authorized devices can decrypt the image and start the system. This prevents unauthorized devices from copying the image.

#### NOTE

For details about the secure boot function, see the *xxxx Secure Boot User Guide* and *Hi35xxVxxx Security Subsystem User Guide*.

#### NOTICE

You are advised to use the secure boot function to reduce product security risks.

## 1.4 Address Space Mapping

Table 1-2 describes the address space mapping.

**Table 1-2** Address space mapping

Start Address	End Address	Function	Size	Description
0x0000_0000	0x0000_7FFF	Pointing to the BootROM	32 KB	
0x0000_8000	0x03FF_FFFF	Reserved	-	



Start Address	End Address	Function	Size	Description
0x0400_0000	0x0400_7FFF	The boot address space is BootROM.	32 KB	
0x0400_8000	0x0401_FFFF	Reserved	-	
0x0402_0000	0x0403_3FFF	The boot address space is BootRAM.	80 KB	
0x0403_4000	0x0EFF_FFFF	Reserved	-	
0x0F00_0000	0x0FFF_FFFF	FMC_MEM space	16 MB	
0x1000_0000	0x1000_03FF	FMC register	1 KB	
0x1000_0400	0x1002_FFFF	Reserved	-	
0x1003_0000	0x1003_0FFF	SDIO0 register	4 KB	
0x1003_1000	0x1003_FFFF	Reserved	-	
0x1004_0000	0x1004_0FFF	SDIO1 register	4 KB	
0x1004_1000	0x101D_FFFF	Reserved	-	
0x101E_0000	0x101E_1FFF	OTP Ctrl register	8 KB	
0x101E_2000	0x101E_FFFF	Reserved	-	
0x101F_0000	0x101F_FFFF	SPACC register	64 KB	
0x1020_0000	0x1025_FFFF	Reserved	-	
0x1026_0000	0x1026_FFFF	IO0_CFG register	64 KB	IO0 pin configuration
0x1027_0000	0x1027_FFFF	Reserved	-	
0x1028_0000	0x1028_0FFF	DMA register	4 KB	
0x1028_1000	0x1028_FFFF	Reserved	-	
0x1029_0000	0x1029_FFFF	ETH register	64 KB	
0x102A_0000	0x102F_FFFF	Reserved	-	
0x1030_0000	0x1030_FFFF	USB2 controller register	64 KB	
0x1031_0000	0x1031_0FFF	USB2 PHY register	4 KB	
0x1031_1000	0x10FF_FFFF	Reserved	-	
0x1100_0000	0x1100_0FFF	TIMER01 register	4 KB	
0x1100_1000	0x1100_1FFF	TIMER23 register	4 KB	



Start Address	End Address	Function	Size	Description
0x1100_2000	0x1100_2FFF	SEC_TIMER01 register	4 KB	
0x1100_3000	0x1100_3FFF	SEC_TIMER23 register	4 KB	
0x1100_4000	0x1100_FFFF	Reserved	-	
0x1101_0000	0x1101_FFFF	CRG register	64 KB	
0x1102_0000	0x1102_8FFF	SYS CTRL register	36 KB	
0x1102_9000	0x1102_9FFF	SVB_PWM register	4 KB	
0x1102_A000	0x1102_AFFF	Tsensor_CTRL register	4 KB	
0x1102_B000	0x1102_BFFF	HPM_CTRL register	4 KB	
0x1102_C000	0x1102_CFFF	STOPWATCH register	4 KB	
0x1102_D000	0x1102_FFFF	Reserved	-	
0x1103_0000	0x1103_0FFF	WDG register	4 KB	
0x1103_1000	0x1103_FFFF	Reserved	-	
0x1104_0000	0x1104_0FFF	UART0 register	4 KB	
0x1104_1000	0x1104_1FFF	UART1 register	4 KB	
0x1104_2000	0x1104_2FFF	UART2 register	4 KB	
0x1104_3000	0x1105_FFFF	Reserved	-	
0x1106_0000	0x1106_0FFF	I2C0 register	4 KB	
0x1106_1000	0x1106_1FFF	I2C1 register	4 KB	
0x1106_2000	0x1106_2FFF	I2C2 register	4 KB	
0x1106_3000	0x1106_FFFF	Reserved	-	
0x1107_0000	0x1107_0FFF	SPI0 register	4 KB	
0x1107_1000	0x1107_1FFF	SPI1 register	4 KB	
0x1107_2000	0x1107_FFFF	Reserved	-	
0x1108_0000	0x1108_0FFF	PWM register	4 KB	
0x1108_1000	0x1108_FFFF	Reserved	-	
0x1109_0000	0x1109_0FFF	GPIO0 register	4 KB	
0x1109_1000	0x1109_1FFF	GPIO1 register	4 KB	
0x1109_2000	0x1109_2FFF	GPIO2 register	4 KB	



Start Address	End Address	Function	Size	Description
0x1109_3000	0x1109_3FFF	GPIO3 register	4 KB	
0x1109_4000	0x1109_4FFF	GPIO4 register	4 KB	
0x1109_5000	0x1109_5FFF	GPIO5 register	4 KB	
0x1109_6000	0x1109_6FFF	GPIO6 register	4 KB	
0x1109_7000	0x1109_7FFF	GPIO7 register	4 KB	
0x1109_8000	0x1109_8FFF	GPIO8 register	4 KB	
0x1109_9000	0x1109_9FFF	GPIO9 register	4 KB	
0x1109_A000	0x1109_AFFF	GPIO10 register	4 KB	
0x1109_B000	0x110F_FFFF	Reserved	-	
0x1110_0000	0x1110_FFFF	LSADC register	64 KB	
0x1111_0000	0x1111_FFFF	RTC register	64 KB	
0x1112_0000	0x1112_FFFF	Reserved	-	
0x1113_0000	0x1113_FFFF	IO1_CFG register	64 KB	IO1 pin configuration
0x1114_0000	0x1115_FFFF	MDDR register	128 KB	
0x1116_0000	0x1116_FFFF	Reserved	-	
0x1117_0000	0x13FF_FFFF	Reserved	-	
0x1400_0000	0x143F_FFFF	NPU register	4 MB	
0x1440_0000	0x170E_FFFF	Reserved	-	
0x170F_0000	0x170F_FFFF	GZIP register	64 KB	
0x1710_0000	0x1713_FFFF	Reserved	-	
0x1714_0000	0x1714_FFFF	VEDU register	64 KB	
0x1715_0000	0x171B_FFFF	Reserved	-	
0x171C_0000	0x171C_FFFF	JPGE register	64 KB	
0x171D_0000	0x1723_FFFF	Reserved	-	
0x1724_0000	0x1724_FFFF	VGS register	64 KB	
0x1725_0000	0x173B_FFFF	Reserved	-	
0x173C_0000	0x173C_FFFF	MIPI RX register	64 KB	
0x173D_0000	0x173F_FFFF	Reserved	-	



Start Address	End Address	Function	Size	Description
0x1740_0000	0x174F_FFFF	VICAP register	1 MB	
0x1750_0000	0x177F_FFFF	Reserved	-	
0x1780_0000	0x1783_FFFF	VIPROC register	256 KB	
0x1784_0000	0x178F_FFFF	Reserved	-	
0x1790_0000	0x1790_FFFF	VPSS register	64 KB	
0x1791_0000	0x1793_FFFF	Reserved	-	
0x1794_0000	0x1794_FFFF	IO2_CFG register	64 KB	IO2 pin configuration
0x1795_0000	0x17BF_FFFF	Reserved	-	
0x17C0_0000	0x17C0_FFFF	AIAO register	64 KB	
0x17C1_0000	0x17C3_FFFF	Reserved	-	
0x17C4_0000	0x17C4_FFFF	Audio Codec register	64 KB	
0x17C5_0000	0x3FFF_FFFF	Reserved	-	
0x4000_0000	0xFFFF_FFFF	DDR address space	3072 MB	



# Contents

<b>2 Hardware.....</b>	<b>2-1</b>
2.1 Package and Pinout.....	2-1
2.1.1 Package.....	2-1
2.1.2 Pinout .....	2-9
2.2 Pin Description.....	2-10
2.3 Soldering Process Recommendations .....	2-10
2.3.1 Requirements on Parameters of the Lead-Free Reflow Soldering Process.....	2-10
2.3.2 Requirements of Mixing Reflow Soldering .....	2-13
2.4 Moisture-Sensitive Specifications .....	2-14
2.4.1 Moisture-Proof Packaging .....	2-14
2.4.2 Storage and Usage .....	2-15
2.4.3 Rebaking.....	2-16
2.5 Electrical Specifications .....	2-17
2.5.1 Power Consumption Parameters.....	2-17
2.5.2 Temperature and Thermal Resistance Parameters .....	2-20
2.5.3 Operating Conditions.....	2-21
2.5.4 Power-On and Power-Off Sequences.....	2-25
2.5.5 DC and AC Electrical Parameters .....	2-25
2.5.6 MIPI/LVDS Rx Electrical Parameters .....	2-32
2.5.7 Audio CODEC Electrical Parameters.....	2-34
2.6 Interface Timings.....	2-36
2.6.1 DDR Interface Timings .....	2-36
2.6.2 SPI FLASH Interface Timings.....	2-38
2.6.3 VI Interface Timings .....	2-41
2.6.4 AIAO Interface Timings.....	2-42
2.6.5 I <sub>2</sub> C Interface Timing.....	2-43
2.6.6 SPI Timings .....	2-44
2.6.7 MIPI RX Timings.....	2-47
2.6.8 SDIO/MMC Interface Timings .....	2-49



## Figures

<b>Figure 2-1</b> Top view .....	2-2
<b>Figure 2-2</b> Bottom view .....	2-3
<b>Figure 2-3</b> Side view .....	2-3
<b>Figure 2-4</b> Enlarged view of detail "A" .....	2-4
<b>Figure 2-5</b> Top view .....	2-6
<b>Figure 2-6</b> Bottom view .....	2-7
<b>Figure 2-7</b> Side view .....	2-7
<b>Figure 2-8</b> Enlarged view of detail "A" .....	2-8
<b>Figure 2-9</b> Curve of the lead-free reflow soldering process .....	2-10
<b>Figure 2-10</b> Measuring the package temperature .....	2-12
<b>Figure 2-11</b> Vacuum packaging materials .....	2-15
<b>Figure 2-12</b> Write timing of CMD/ADDR relative to CK for the DDR3/3L SDRAM .....	2-36
<b>Figure 2-13</b> Write timing of DQS relative to CK for the DDR3/3L SDRAM .....	2-36
<b>Figure 2-14</b> Read timings of CK relative to DQS for the DDR3/3L SDRAM .....	2-37
<b>Figure 2-15</b> Read timings of DQS relative to DQ for the DDR3/3L SDRAM .....	2-37
<b>Figure 2-16</b> SPI FLASH input timing (SDR mode) .....	2-39
<b>Figure 2-17</b> SPI FLASH input timing (DDR mode) .....	2-39
<b>Figure 2-18</b> SPI FLASH output timing (SDR mode) .....	2-40
<b>Figure 2-19</b> SPI FLASH output timing (DDR mode) .....	2-40
<b>Figure 2-20</b> VI interface timings in CMOS mode .....	2-41
<b>Figure 2-21</b> RX timing of the I <sub>2</sub> S interface .....	2-42
<b>Figure 2-22</b> TX timing of the I <sub>2</sub> S interface .....	2-42
<b>Figure 2-23</b> RX timing of the PCM interface .....	2-43
<b>Figure 2-24</b> TX timing of the PCM interface .....	2-43
<b>Figure 2-25</b> I <sub>2</sub> C transfer timing .....	2-44



<b>Figure 2-26</b> SPICK timing.....	2-45
<b>Figure 2-27</b> SPI timing in master mode (sph = 1).....	2-45
<b>Figure 2-28</b> SPI timing in master mode (sph = 0).....	2-45
<b>Figure 2-29</b> Clock timings of the MIPI RX DPHY/Sub-lvds/LVDS/HiSPi interface.....	2-48
<b>Figure 2-30</b> CMD/DATA input/output timings of the eMMC in DS/HS mode.....	2-49
<b>Figure 2-31</b> CMD/DATA input/output timings of the eMMC in HS200 mode.....	2-51
<b>Figure 2-32</b> Data output timing in HS400 mode.....	2-52
<b>Figure 2-33</b> Data input timing in HS400 mode.....	2-53
<b>Figure 2-34</b> CMD/DATA output timings of the SDIO in DS mode .....	2-54
<b>Figure 2-35</b> CMD/DATA input timings of the SDIO in DS mode .....	2-54
<b>Figure 2-36</b> CMD/DATA output timings of the SDIO in HS mode .....	2-55
<b>Figure 2-37</b> CMD/DATA input timings of the SDIO in HS mode .....	2-55



## Tables

<b>Table 2-1</b> Package dimensions .....	2-4
<b>Table 2-2</b> Package dimensions .....	2-8
<b>Table 2-3</b> Pin quantity.....	2-9
<b>Table 2-4</b> Parameters of the lead-free reflow soldering process.....	2-11
<b>Table 2-5</b> Temperature resistance standard for the lead-free package according to IPC/JEDEC 020D .....	2-12
<b>Table 2-6</b> Mixing reflow soldering parameters .....	2-13
<b>Table 2-7</b> Thermal resistance standard for the lead package .....	2-14
<b>Table 2-8</b> Floor life.....	2-16
<b>Table 2-9</b> Rebaking reference .....	2-16
<b>Table 2-10</b> For more power consumption scenarios, see the Hi35xxVxxx Power Consumption Test Report.....	2-18
<b>Table 2-11</b> For more power consumption scenarios, see the Hi35xxVxxx Power Consumption Test Report.....	2-19
<b>Table 2-12</b> For more power consumption scenarios, see the Hi35xxVxxx Power Consumption Test Report.....	2-20
<b>Table 2-13</b> Operating environment parameters of Hi3516CV610 .....	2-20
<b>Table 2-14</b> Requirements on the junction temperatures of Hi3516CV610 .....	2-21
<b>Table 2-15</b> Thermal resistance parameters .....	2-21
<b>Table 2-16</b> Operating condition of the core power supply.....	2-22
<b>Table 2-17</b> Operating conditions for the power supplies of Hi3516CV610-10B/20B/20S/20G under normal voltage .....	2-22
<b>Table 2-18</b> Operating conditions for the power supplies of Hi3516CV610-00B/00S/00G under normal voltage .....	2-23
<b>Table 2-19</b> Destructive voltage values of Hi3516CV610-10B/20B/20S/20G.....	2-24
<b>Table 2-20</b> Destructive voltage values of Hi3516CV610-00B/00S/00G .....	2-24



<b>Table 2-21</b> DC electrical parameters (DVDD33/DVDD3318_Sensor/DVDD_3318(BGA)/DVDD3318_FLASH=3.3V).....	2-25
<b>Table 2-22</b> DC electrical parameters (DVDD3318_FLASH/DVDD3318_Sensor/DVDD3318(BGA)=1.8V).....	2-29
<b>Table 2-23</b> Differential DC electrical parameters .....	2-32
<b>Table 2-24</b> MIPI DPHY high-speed (HS) DC parameters.....	2-33
<b>Table 2-25</b> MIPI DPHY high-speed (HS) AC parameters .....	2-33
<b>Table 2-26</b> MIPI DPHY low-power (LP) DC parameters .....	2-34
<b>Table 2-27</b> MIPI DPHY low-power (LP) AC parameters.....	2-34
<b>Table 2-28</b> Overall specifications.....	2-34
<b>Table 2-29</b> Major DAC specifications .....	2-35
<b>Table 2-30</b> Major ADC specifications .....	2-35
<b>Table 2-31</b> Major MICBIAS specifications .....	2-35
<b>Table 2-32</b> Major MICPGA specifications.....	2-35
<b>Table 2-33</b> CK parameters of the DDR3 SDRAM (DDR3/3L-2133).....	2-38
<b>Table 2-34</b> DQS parameters of the DDR3 SDRAM (DDR3/3L-2133) .....	2-38
<b>Table 2-35</b> SPI flash input timing parameters (voltage = 3.3 V) .....	2-39
<b>Table 2-36</b> SPI flash input timing parameters (voltage = 1.8 V) .....	2-40
<b>Table 2-37</b> SPI flash output timing parameters (voltage = 3.3 V).....	2-40
<b>Table 2-38</b> SPI flash output timing parameters (voltage = 1.8 V).....	2-41
<b>Table 2-39</b> VI interface timing parameters.....	2-42
<b>Table 2-40</b> Timing parameters of the I <sub>2</sub> S interface.....	2-42
<b>Table 2-41</b> Timing parameters of the PCM interface.....	2-43
<b>Table 2-42</b> Timing parameters of the I <sub>2</sub> C interface .....	2-44
<b>Table 2-43</b> SPI timing parameters.....	2-46
<b>Table 2-44</b> MIPI RX DPHY/Sub-lvds/LVDS/HiSPi timing parameters .....	2-48
<b>Table 2-45</b> CMD/DATA input/output timing parameters of the eMMC in DS mode.....	2-49
<b>Table 2-46</b> CMD/DATA input/output timing parameters of the eMMC in HS mode .....	2-50
<b>Table 2-47</b> CMD/DATA input/output timing parameters in HS200 mode .....	2-51
<b>Table 2-48</b> Data output timing parameters in HS400 mode .....	2-52
<b>Table 2-49</b> Data input timing parameters in HS400 mode .....	2-53
<b>Table 2-50</b> Input/output timings of the SDIO in DS mode.....	2-55



**Table 2-51 CMD/DATA input/output timing parameters of the SDIO in HS mode ..... 2-56**



# 2 Hardware

## 2.1 Package and Pinout

### 2.1.1 Package

---

#### NOTICE

Hi3516CV610-10B/20B/20S/20G uses the QFN package, while Hi3516CV610-00B/00S/00G uses the TFBGA package.

---

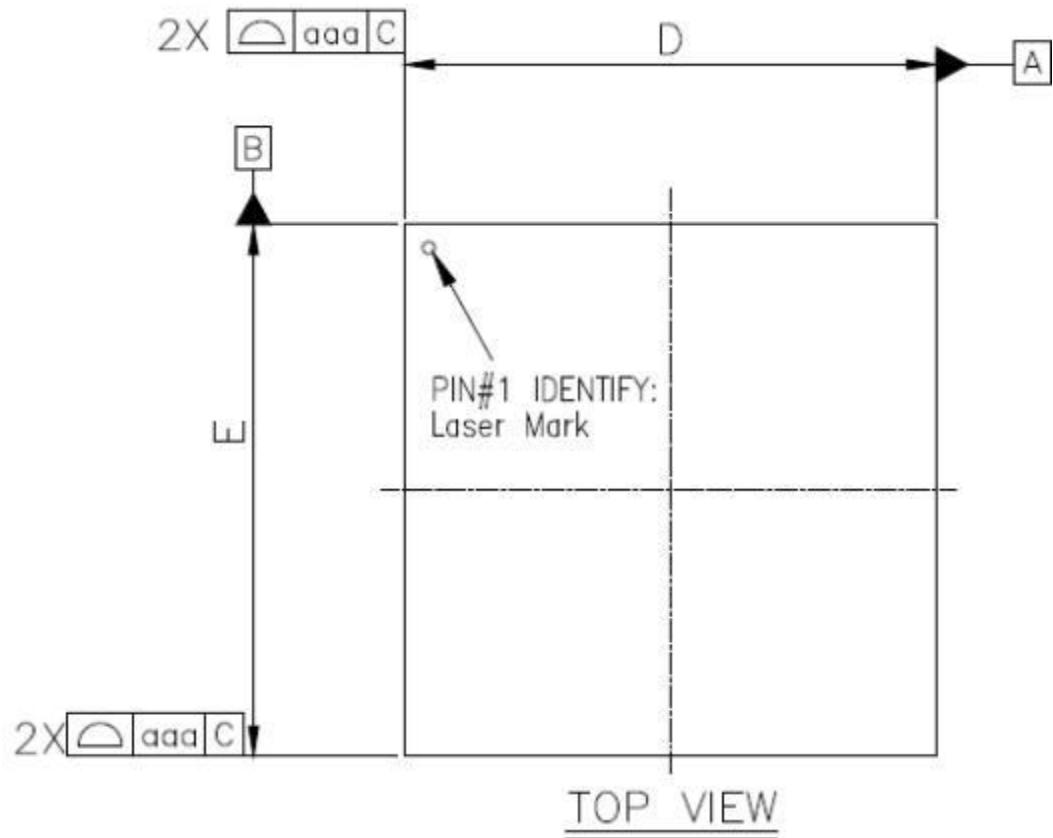
#### Hi3516CV610-10B/20B/20S/20G

The chip uses the quad flat no-lead (QFN) package. It has 88 pins, its body size is 9 mm x 9 mm (0.35 in. x 0.35 in.), and its ball pitch is 0.35 mm (0.03 in.).

[Figure 2-1](#)to [Figure 2-4](#)show the package of Hi3516CV610-10B/20B/20S/20G. [Table 2-1](#)shows the package dimensions.

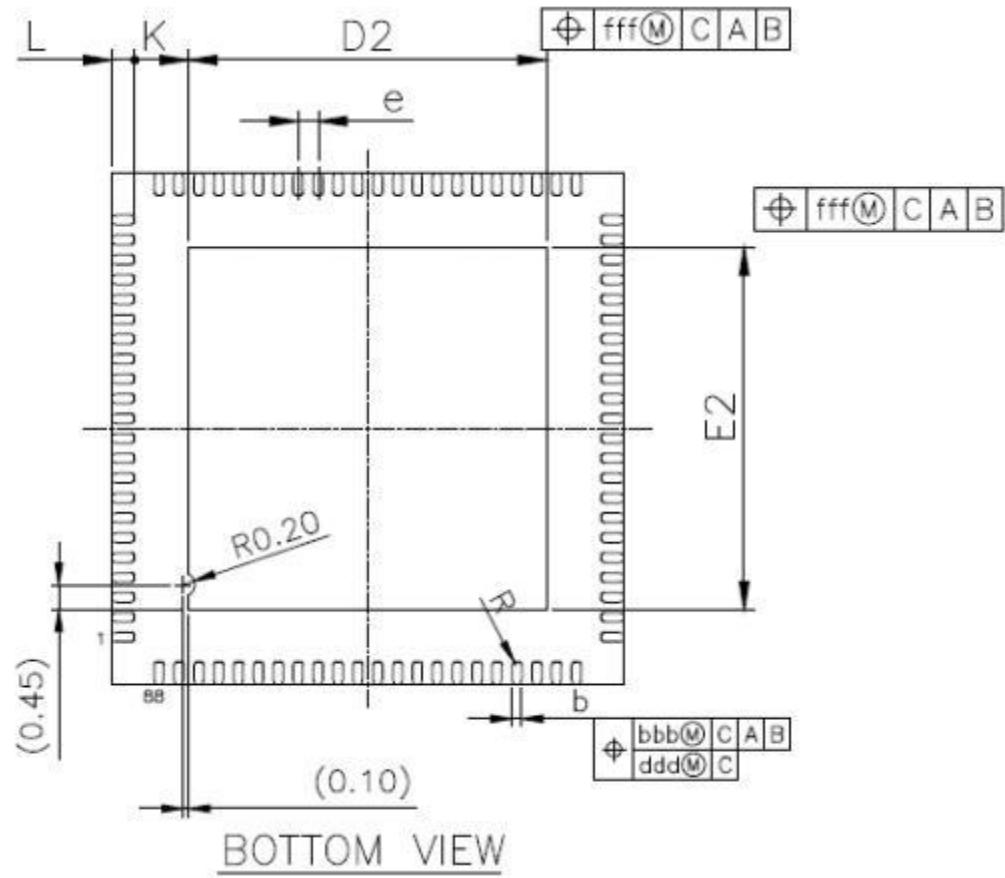


**Figure 2-1** Top view





**Figure 2-2** Bottom view



**Figure 2-3** Side view

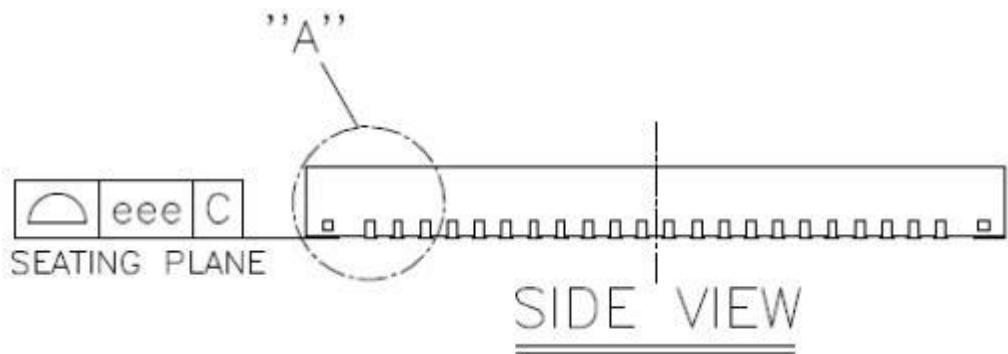
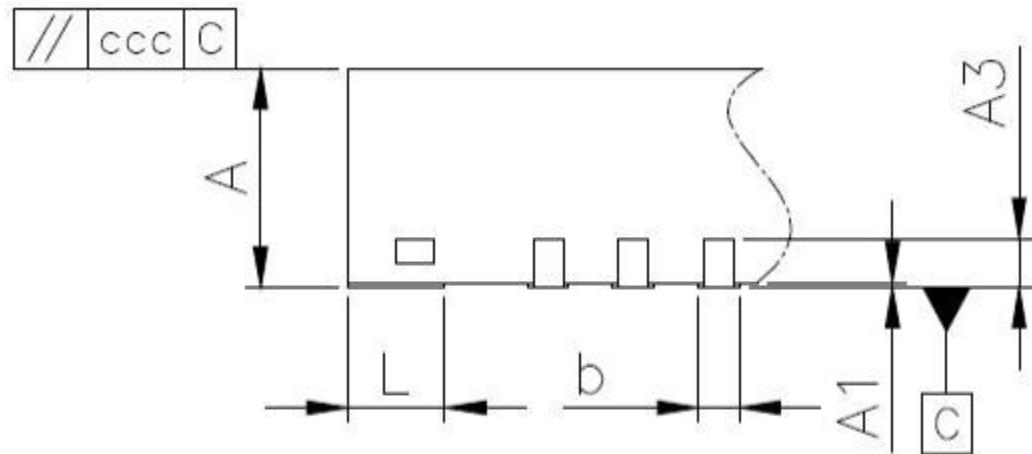




Figure 2-4 Enlarged view of detail "A"



DETAIL : "A"

Table 2-1 Package dimensions

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	0.85	0.90	0.95	0.033	0.035	0.037
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.20 REF			0.008 REF		
b	0.12	0.17	0.22	0.005	0.007	0.009
D	8.90	9.00	9.10	0.350	0.354	0.358
E	8.90	9.00	9.10	0.350	0.354	0.358
D2	6.20	6.30	6.40	0.244	0.248	0.252
E2	6.30	6.40	6.50	0.248	0.252	0.256
e	0.35 BSC			0.014 BSC		
L	0.30	0.40	0.50	0.012	0.016	0.020
K	0.20	---	---	0.008	---	---
R	0.06	---	0.11	0.002	---	0.004
aaa	0.10			0.004		



Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
bbb	0.07			0.003		
ccc	0.10			0.004		
ddd	0.05			0.002		
eee	0.08			0.003		
fff	0.10			0.004		

#### NOTE

- Controlling dimension: millimeter.
- Reference document: JEDEC MO-220.

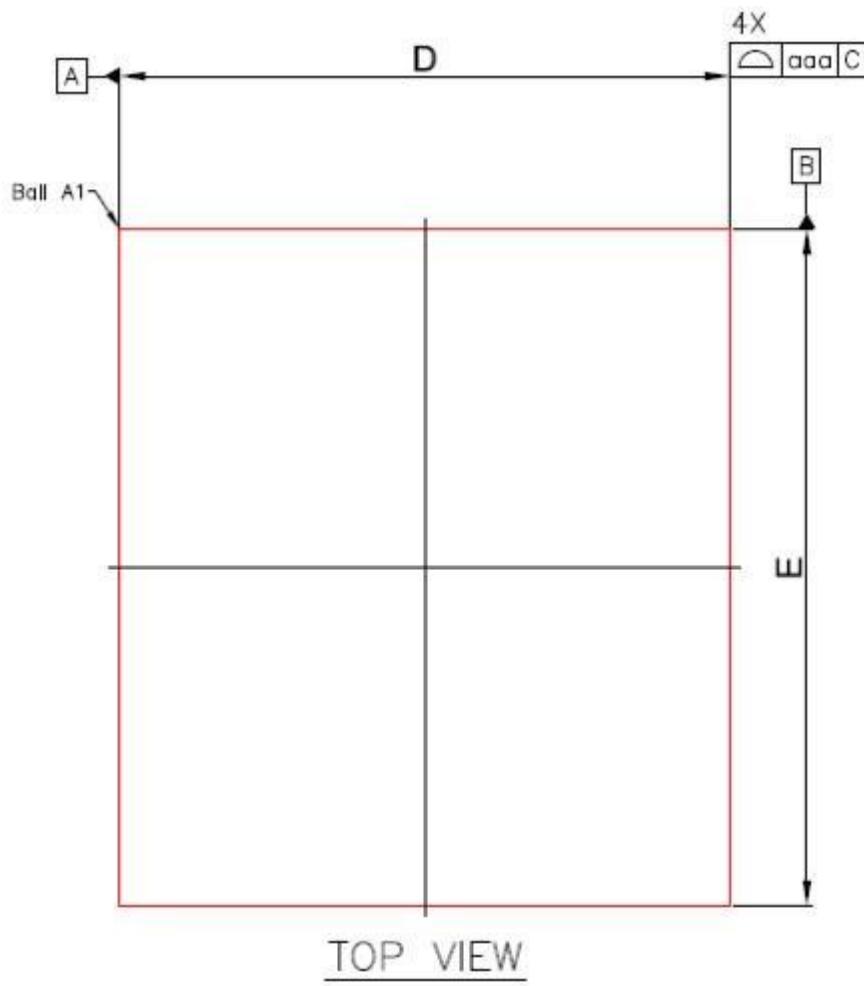
### Hi3516CV610-00B/00S/00G

The chip uses the thin and fine pitch ball grid array (TFBGA) package. It has 289 pins, its body size is 12 mm x 13.3 mm (0.47 in. x 0.52 in.), and its ball pitch is 0.65 mm (0.03 in.).

[Figure 2-5](#)to [Figure 2-8](#)show the package of Hi3516CV610-00B/00S/00G.[Table 2-2](#)shows the package dimensions.

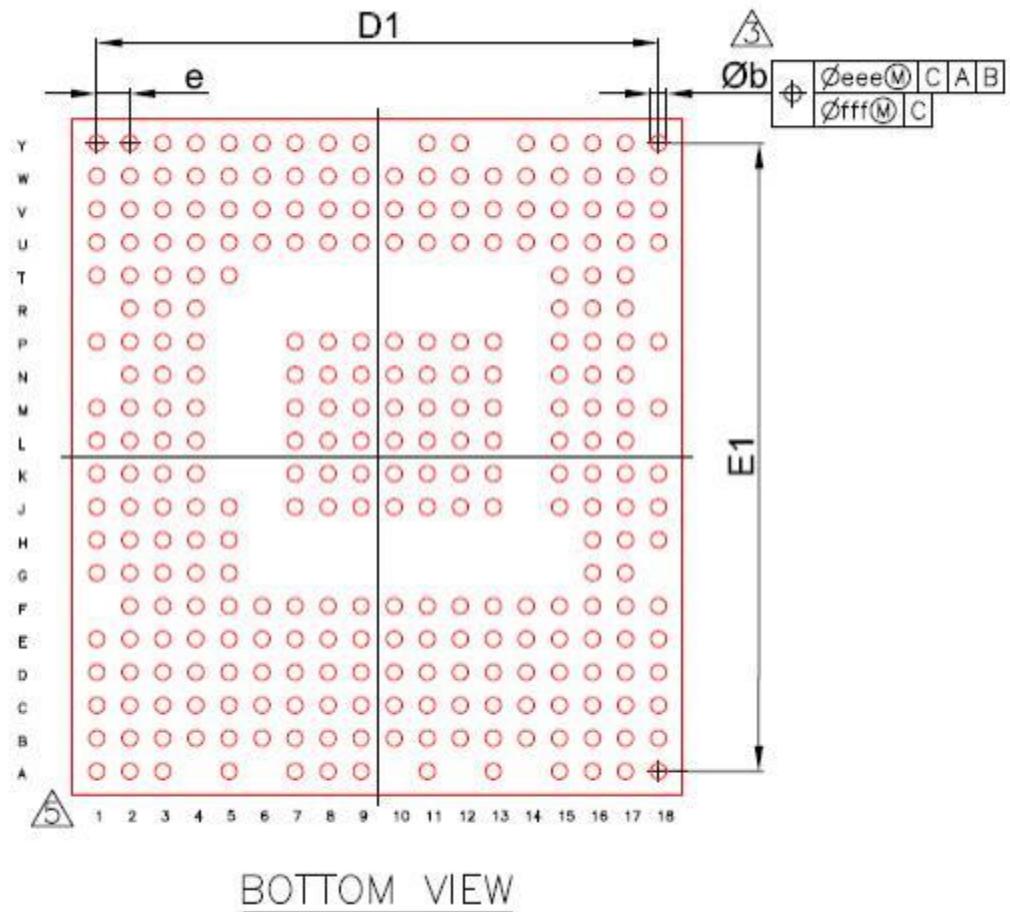


**Figure 2-5** Top view





**Figure 2-6** Bottom view



**Figure 2-7** Side view

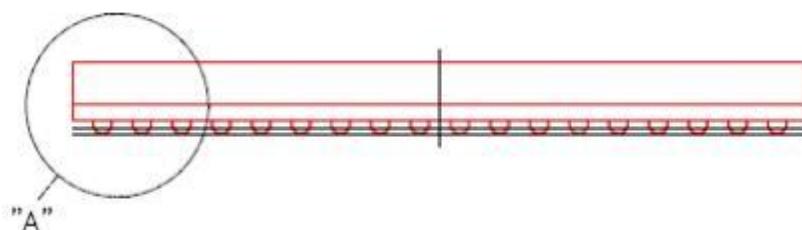




Figure 2-8 Enlarged view of detail "A"

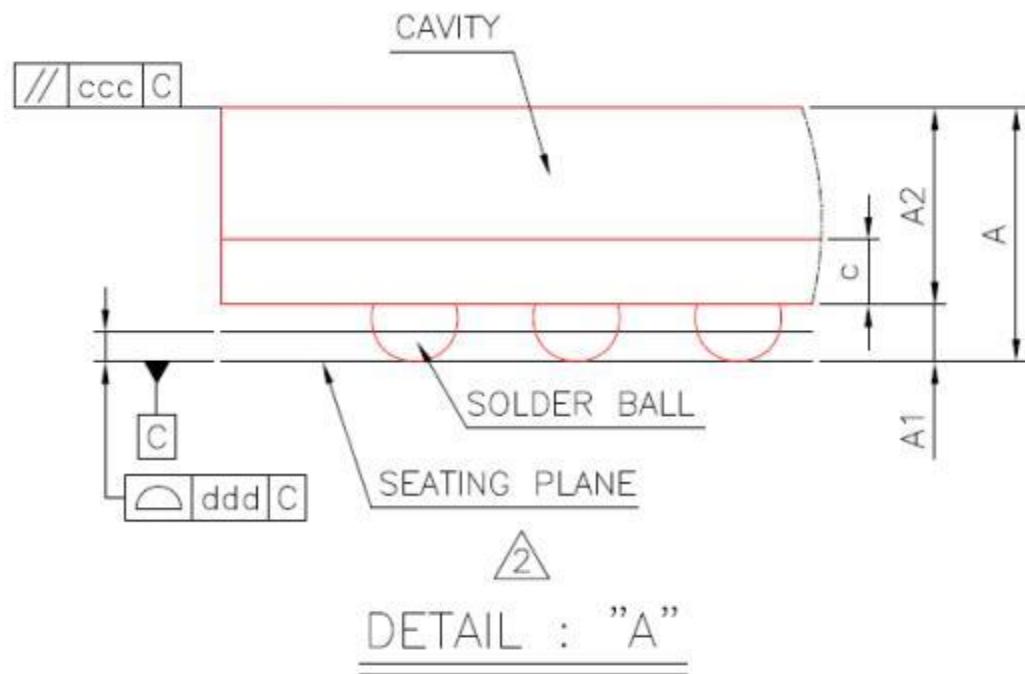


Table 2-2 Package dimensions

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.100	1.170	1.240	0.043	0.046	0.049
A1	0.160	0.210	0.260	0.006	0.008	0.010
A2	0.910	0.960	1.010	0.036	0.038	0.040
c	0.220	0.260	0.300	0.009	0.010	0.011
D	11.900	12.000	12.100	0.469	0.472	0.476
E	13.200	13.300	13.400	0.520	0.524	0.528
D1	---	11.050	---	---	0.435	---
E1	---	12.350	---	---	0.486	---
e	---	0.650	---	---	0.026	---
b	0.260	0.310	0.360	0.010	0.012	0.014
aaa	0.150			0.006		
ccc	0.150			0.006		



Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
ddd	0.080			0.003		
eee	0.150			0.006		
fff	0.080			0.003		
MD/ME	18/20					

### NOTE

- Controlling dimension: millimeter.
- Primary datum c and seating plane are defined by the spherical crowns of the solder balls.
- Dimension b is measured at the maximum solder ball diameter, parallel to primary datum c.
- Special characteristics c class: ccc, ddd.
- The pattern of pin 1 fiducial is for reference only.
- Ball placement use 0.30 mm solder ball. Bga pad solder mask opening = 0.275 mm.

## 2.1.2 Pinout

Hi3516CV610-10B/20B/20S/20G has 88 pins, while Hi3516CV610-00B/00S/00G has 289 pins. [Table 2-3](#) lists the pin quantity of the Hi3516CV610 by type.

**Table 2-3** Pin quantity

Pin Type	Quantity	
	Hi3516CV610-10B/20B/20S/20G	Hi3516CV610-00B/00S/00G
I/O	68	159
Digital power	9	20
Digital GND	1	82
Others/Analog power	6	5
Others/Analog GND	1	15
DDR reference power (VREF)	3	8
Total	88	289



### NOTE

The exposed thermal pad (Epad) connects to VSS (digital GND) and is not included in [Table 2-3](#). This is applicable only to Hi3516CV610-10B/20B/20S/20G.

## 2.2 Pin Description

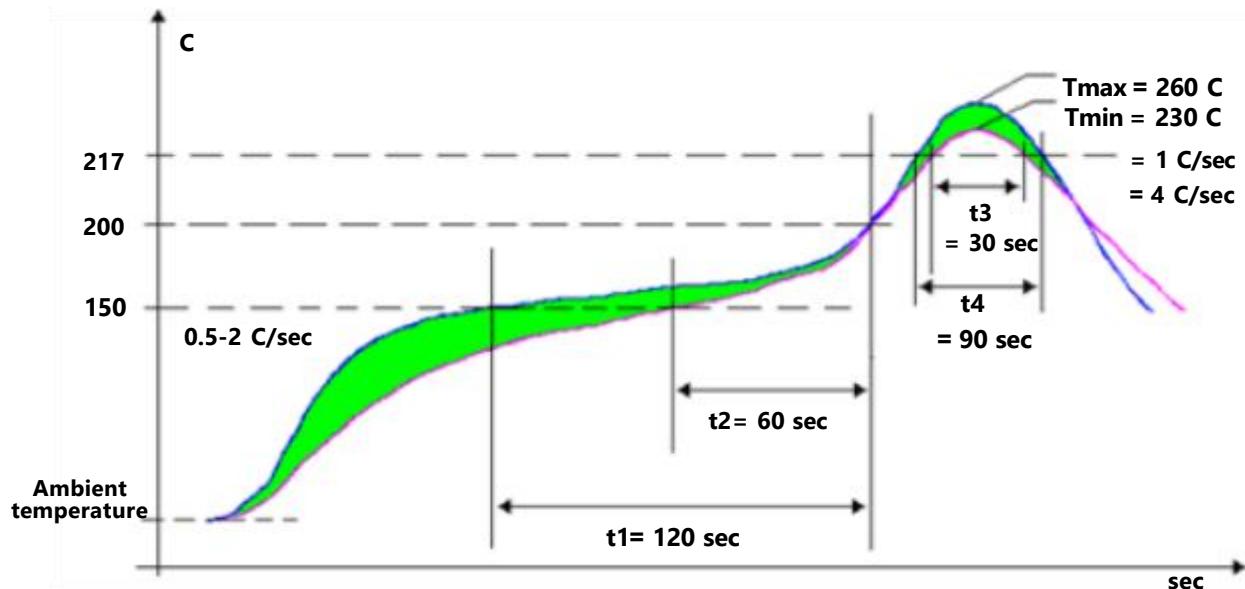
For details about the pins, power domain of the pins, and default status of digital pins, see the *Hi35xxVxxx\_PINOUT\_EN*.

## 2.3 Soldering Process Recommendations

### 2.3.1 Requirements on Parameters of the Lead-Free Reflow Soldering Process

[Figure 2-9](#) shows the curve of the lead-free reflow soldering process.

**Figure 2-9** Curve of the lead-free reflow soldering process



[Table 2-4](#) describes parameters of the lead-free reflow soldering process.



**Table 2-4** Parameters of the lead-free reflow soldering process

Zone	Duration	Heating Rate	Peak Temperature	Cooling Rate
Preheat zone (40°C–150°C or 104°F–302°F)	60s–150s	≤ 2.0°C/s (≤ 36°F/s)	-	-
Uniform temperature zone (150°C–200°C or 302°F–392°F)	60s–120s	< 1.0°C/s (< 34°F/s)	-	-
Reflow zone (greater than 217°C or 423°F)	30s–90s	-	230°C–260°C (446°F–500°F )	-
Cooling zone (Tmax to 180°C or 356°F)	-	-	-	1.0°C/s (34°F/s) ≤ Slope ≤ 4.0°C/s (39°F/s)

#### NOTE

- Preheat zone: The temperature range is 40°C–150°C (104°F–302°F), the heating rate is about 2°C/s (36°F/s), and the duration of this temperature zone is 60s–150s.
- Uniform temperature zone: The temperature range is 150°C–200°C (302°F–392°F), the temperature is increased steadily, the heating rate is less than 1°C/s (34°F/s), and the zone duration is 60s–120s. (Note: Slow heating is required for this zone. Otherwise, improper soldering easily occurs.)
- Reflow zone: The temperature increases from 217°C (423°F) to Tmax, and then decreases from Tmax to 217°C (423°F). The zone duration is 30s–90s.
- Cooling zone: The temperature decreases from Tmax to 180°C (356°F). The maximum cooling rate is 4°C/s (39°F/s).
- It should take no more than 6 minutes for the temperature to increase from the 25°C (77°F) ambient temperature to 250°C (482°F).
- The values on the reflow soldering curve shown in [Figure 2-9](#) are recommended values. The values need to be adjusted on the client as required.
- Typically, the duration of the reflow zone is 60s–90s. For the boards with great heat capacity, the duration can be prolonged to 120s. For details about the requirements on package thermal resistance, see the IPC/JEDEC J-STD-020D standard. For details about the method of measuring the package temperature, see the JEP 140 standard.

[Table 2-5](#) describes the temperature resistance standard for the lead-free package according to IPC/JEDEC 020D.



**Table 2-5** Temperature resistance standard for the lead-free package according to IPC/JEDEC 020D

Package Thickness	Temperature 1 (Package Volume < 350 mm <sup>3</sup> or 0.02 in. <sup>3</sup> )	Temperature 2 (Package Volume = 350-2000 mm <sup>3</sup> or 0.02-0.12 in. <sup>3</sup> )	Temperature 3 (Package Volume > 2000 mm <sup>3</sup> or 0.12 in. <sup>3</sup> )
< 1.6 mm (0.06 in.)	260°C (500°F)	260°C (500°F)	260°C (500°F)
1.6 mm to 2.5 mm (0.06 in. to 0.10 in.)	260°C (500°F)	250°C (482°F)	245°C (473°F)
> 2.5 mm (0.10 in.)	250°C (482°F)	245°C (473°F)	245°C (473°F)

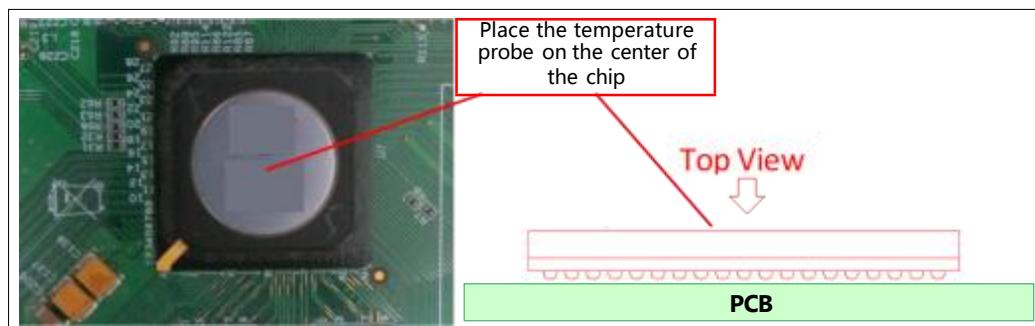
The component soldering terminals (such as the solder ball and pin) and external heat sinks are not considered for volume calculation.

The method of measuring the reflow soldering process curve is as follows:

According to the JEP140 standard, to measure the package temperature, you are advised to place the temperature measuring probe of the thermocouple close to the chip surface if the chip package is thin, or to drill a hole on the package surface and place the temperature measuring probe of the thermocouple into the hole if the chip package is thick. The second method is recommended for all components because of the requirement on quantizing the component thickness.

However, this method is not applicable if the chip package is too thin to drill a hole. See [Figure 2-10](#).

**Figure 2-10** Measuring the package temperature





## 2.3.2 Requirements of Mixing Reflow Soldering

Lead-free components must be properly soldered during mixing reflow soldering. [Table 2-6](#)describes mixing reflow soldering parameters

**Table 2-6** Mixing reflow soldering parameters

Zone		Lead BGA	Lead-free BGA	Other Components
Preheat zone (40°C–150°C or 104°F–302°F)	Duration	60s–150s		
	Heating up slope	< 2.5°C/s (37°F/s)		
Uniform temperature zone (150°C–183°C or 302°F–361°F)	Duration	30–90s		
	Heating up slope	< 1.0°C/s (34°F/s)		
Reflow zone (greater than 183°C or 361°F)	Peak temperature	210°C– 240°C (410°F– 464°F)	220°C– 240°C (428°F– 464°F)	210°C–245°C (410°F–473°F)
	Duration	30s–120s	60s–120s	30s–120s
Cooling zone (Tmax to 150°C or 302°F)	Cooling down slope	1.0°C/s (34°F/s) ≤ Slope ≤ 4.0°C/s (39°F/s)		

### NOTICE

The preceding parameter values are provided based on the soldering joint temperature. The maximum and minimum soldering joint temperatures for the board must meet the requirements described in the preceding table.

When the soldering curve is adjusted, the package thermal resistance requirements on the board components must be met. For details about the requirements on package thermal resistance, see the IPC/JEDEC J-STD-020D standard. For details about the method of measuring the package temperature, see the JEP 140 standard.

[Table 2-7](#)describes the thermal resistance standard for the lead package according to the IPC/JEDEC 020D standard.



**Table 2-7** Thermal resistance standard for the lead package

Package Thickness	Temperature 1 (Package Volume < 350 mm <sup>3</sup> or 0.02 in. <sup>3</sup> )	Temperature 1 (Package Volume ≥ 350 mm <sup>3</sup> or 0.02 in. <sup>3</sup> )
< 2.5 mm (0.10 in.)	235°C (455°F)	220°C (428°F)
≥ 2.5 mm (0.10 in.)	220°C (428°F)	220°C (428°F)

The component soldering terminals (such as the solder ball and pin) and external heat sinks are not considered for volume calculation.

According to the JEP 140 standard, the method of measuring the temperature of the package soldered with the mixing technology is the same as that for measuring the temperature of the package soldered with the lead-free technology. For details, see section [2.3.1 "Requirements on Parameters of the Lead-Free Reflow Soldering Process."](#)

## 2.4 Moisture-Sensitive Specifications

This chapter defines the usage principles for moisture-sensitive ICs to ensure that ICs are properly used. Related terms are explained as follows:

- Floor life: longest period during which the chip can be stored in the workshop below 30°C (86°F) and 60% relative humidity (RH), that is, the time from moisture barrier bag (MBB) unpacking to reflow soldering
- Desiccant: a material for absorbing moisture and keeping the product dry
- Humidity indicator card (HIC): a card that indicates the humidity status
- Moisture sensitivity level (MSL): a level for measuring the moisture degree.
- MBB: a vacuum bag for protecting products against moisture
- Solder reflow: reflow soldering
- Shelf life: normal storage period of a product with the MBB [MSL]

The MSL of this product is 3.

### 2.4.1 Moisture-Proof Packaging

#### 2.4.1.1 Basic Information

The vacuum packaging materials include:

- An HIC
- An MBB

- Desiccant

**Figure 2-11 Vacuum packaging materials**



#### 2.4.1.2 Incoming Inspection

When the vacuum bag is unpacked before SMT in the factories of customers or outsourcing vendors:

- If the largest indicator dot of the HIC is not in blue or khaki, rebake the product by referring to [Table 2-9](#).
- If the 10% RH dot of the HIC is in blue or khaki, the product is dry. In this case, replace the desiccant and pack the product into a vacuum bag.
- If the 10% RH dot of the HIC is not in blue or khaki and the 5% RH dot is in red or light green, the product has become damp. In this case, rebake the product according to [Table 2-9](#).

#### 2.4.2 Storage and Usage

##### [Storage Environment]

You are advised to use vacuum packaging for the product and store it below 30°C (86°F) and 60% RH.

##### [Shelf Life]

Normal storage period of a product with the MBB

Below 30°C (86°F) and 60% RH, the shelf life of the product with vacuum packaging is not less than 12 months.

##### [Floor Life]



[Table 2-8](#)describes the floor life below 30°C (86°F) and 60% RH.

**Table 2-8** Floor life

MSL	Floor life (out of bag) at factory ambient $\leq 30^{\circ}\text{C}/60\%$ RH or as stated
1	Unlimited at $\leq 30^{\circ}\text{C}/85\%$ RH
2	1 year
2a	4 weeks
3	168 hours
4	72 hours
5	48 hours
5a	24 hours
6	Mandatory bake before use, must be reflowed within the time limit specified on the label

[Usage of Moisture-Sensitive Products]

- If the product has been exposed to air for more than 2 hours at 30°C (86°F) or lower and at most 60% RH, rebake it and pack it into a vacuum bag.
- If the product has been exposed to air for no more than 2 hours at 30°C (86°F) or lower and at most 60% RH, replace the desiccant and pack the chip into a vacuum bag.

For details about other storage and usage rules, see the JEDEC J-STD-033A standard.

### 2.4.3 Rebaking

Application scope: all moisture-sensitive ICs that need to be rebaked.

[Table 2-9](#) lists the rebaking reference.

**Table 2-9** Rebaking reference

Body Thickness	Level	Bake@125°C	Bake@90°C $\leq 5\%$ RH	Bake@40°C $\leq 5\%$ RH
$\leq 1.4\text{ mm}$	2a	3 hours	11 hours	5 days
	3	7 hours	23 hours	9 days
	4	7 hours	23 hours	9 days



Body Thickness	Level	Bake@125°C	Bake@90°C ≤ 5% RH	Bake@40°C ≤ 5% RH
≤ 2.0 mm	5	7 hours	24 hours	10 days
	5a	10 hours	24 hours	10 days
	2a	16 hours	2 days	22 days
	3	17 hours	2 days	23 days
	4	20 hours	3 days	28 days
	5	25 hours	4 days	35 days
	5a	40 hours	6 days	56 days
	2a	48 hours	7 days	67 days
	3	48 hours	8 days	67 days
	4	48 hours	10 days	67 days
	5	48 hours	10 days	67 days
	5a	48 hours	10 days	67 days

### NOTE

- Table 2-9 lists the minimum rebaking time required for damp products.
- Low-temperature rebaking is recommended.
- For details, see the JEDEC standard.

## 2.5 Electrical Specifications

### 2.5.1 Power Consumption Parameters

- Typical Scenarios of Hi3516CV610-10B: Monocular 4M@25 fps Linear Mode
  - Configuration:**
    - CPU: dual-core Cortex-A7 950 MHz
    - DDR: built-in DDR2, 1.8 V power supply, 1333 Mbps
  - Service description:**
    - **MPP:** The VI captures 1-channel 2560 x 1440@25 fps input under normal illumination, FE-BE is online, and 3DNR (with frame compression) is enabled for the VI. The VI-VPSS sends 2560 x 1440@25 fps data to VPSS group 0 online (in interrupt START mode). VPSS group 0 (overlaid four Cover regions) outputs multi-channel data to the VENC and SVP.



Specifically, VPSS channel 0 (with wrapping enabled) outputs data to the VENC for encoding {large stream 2560 x 1440@25 fps 4 Mbps/H.265/NormalP}. VPSS channel 1 outputs data to the VENC for encoding {small stream 720 x 480@25 fps 512 kbps/H.265/NormalP}. VPSS channel 2 outputs 1024 x 576@10 fps data to the SVP module for processing.

- **SVP:** Run the object detection model to process 1-channel 1024 x 576@10 fps data.
- **Network:** Play large streams on demand.
- **Audio:** One channel for capturing 16 kHz mono and one channel for encoding and decoding; one channel for outputting 16 kHz mono and one channel for AI with AGC/ANR/AEC enabled.

**Table 2-10** For more power consumption scenarios, see the Hi35xxVxxx Power Consumption Test Report.

Chip Model	Junction Temperature (°C)	VDD (mW)	DDRIO_1.8V (mW)	SOC_1.8 V (mW)	SOC_3.3 V (mW)	Total Power Consumption (mW)
Hi3516CV610 -10B	37.2	313	196	14	194	717
	105	546	241	14	216	1017

- Typical Scenarios of Hi3516CV610-20S: Monocular 4M@25 fps Linear Mode
  - Configuration:**
    - CPU: dual-core Cortex-A7 950 MHz
    - DDR: built-in DDR3L, 1.35 V power supply, 2133 Mbps
  - Service description:**
    - **MPP:** The VI captures 1-channel 2560 x 1440@25 fps input under normal illumination, FE-BE is online, and 3DNR (with frame compression) is enabled for the VI. The VI-VPSS sends 2560 x 1440@25 fps data to VPSS group 0 online (in interrupt START mode). VPSS group 0 (overlaying four Cover regions) outputs multi-channel data to the VENC and SVP. Specifically, VPSS channel 0 (with compact compression and wrapping enabled) outputs data to the VENC for encoding {large stream 2560 x 1440@25 fps 4 Mbps/H.265/NormalP} + {2560 x 1440@1 fps/JPEG}. VPSS channel 1 outputs data to the VENC for encoding {small stream 720 x 480@25 fps 512 kbps/H.265/NormalP}. VPSS channel 2 outputs 1024 x 576@10 fps YUV data.
    - **SVP:** Run the yolov3\_tiny model to cyclically process 1-channel 416 x 416 image without frame rate limitation.
    - **Network:** Play large streams on demand.



- **Audio:** One channel for capturing 16 kHz mono and one channel for encoding and decoding; one channel for outputting 16 kHz mono and one channel for AI with AGC/ANR/AEC enabled.

**Table 2-11** For more power consumption scenarios, see the Hi35xxVxxx Power Consumption Test Report.

Chip Model	Junction Temperature (°C)	VDD (mW)	DDRIO_1.35V (mW)	SOC_1.8V (mW)	SOC_3.3V (mW)	Total Power Consumption (mW)
Hi3516CV610-20S	40.6	539	304	14	186	1043
	105	760	337	13	221	1331

- Typical Scenarios of Hi3516CV610-00S: Monocular 4M@25 fps Linear Mode
  - Configuration:**
    - CPU: dual-core Cortex-A7 950 MHz
    - DDR: built-in DDR3L, 1.5 V power supply, 2133 Mbps
  - Service description:**
    - **MPP:** The VI captures 1-channel 2560 x 1440@25 fps input under normal illumination, FE-BE is online, and 3DNR (with frame compression) is enabled for the VI. The VI-VPSS sends 2560 x 1440@25 fps data to VPSS group 0 online (in interrupt START mode). VPSS group 0 (overlaying four Cover regions) outputs multi-channel data to the VENC and SVP. Specifically, VPSS channel 0 (with compact compression and wrapping enabled) outputs data to the VENC for encoding {large stream 2560 x 1440@25 fps 4 Mbps/H.265/NormalP} + {2560 x 1440@1 fps/JPEG}. VPSS channel 1 outputs data to the VENC for encoding {small stream 720 x 480@25 fps 512 kbps/H.265/NormalP}. VPSS channel 2 outputs 1024 x 576@10 fps YUV data.
    - **SVP:** Run the yolov3\_tiny model to cyclically process 1-channel 416 x 416 image without frame rate limitation.
    - **Network:** Play large streams on demand.
    - **Audio:** One channel for capturing 16 kHz mono and one channel for encoding and decoding; one channel for outputting 16 kHz mono and one channel for AI with AGC/ANR/AEC enabled.



**Table 2-12** For more power consumption scenarios, see the Hi35xxVxxx Power Consumption Test Report.

Chip Model	Junction Temperature (°C)	VDD (mW)	DDRIO_1.5V (mW)	SOC_1.8V (mW)	SOC_3.3V (mW)	Total Power Consumption (mW)
Hi3516CV610-00S	44	503	132	11	183	829
	105	723	186	13	191	1113

### NOTICE

- The power consumption parameters are based on the typical working scenarios of the Hi3516CV610-10/20/00 series chips.
- Design board power supplies by following the *Hi35xxVxxx Hardware Design User Guide*.

## 2.5.2 Temperature and Thermal Resistance Parameters

Table 2-13, Table 2-14, and Table 2-15 describe temperature and thermal resistance parameters.

### NOTE

- The thermal resistance is provided in compliance with the JEDEC JESD51 series of standards. The actual system design and environment may be different.
  - For details about  $\theta_{JA}$ , see the JEDEC Standard No.51-2.
  - For details about  $\theta_{JB}$ , see the JEDEC Standard No.51-8.
  - For details about  $\theta_{JC}$ , see the following standards:
    - 1) MIL-STD-883 1012.1
    - 2) SEMI G30-88
- The chip junction temperature is proportional to the chip power consumption. Ensure that the junction temperature is appropriate to match power supplies.

**Table 2-13** Operating environment parameters of Hi3516CV610

Chip	Ambient Temperature for Working ( $T_A$ )	
	Min (°C)	Max (°C)
Hi3516CV610	-20	70



**Table 2-14** Requirements on the junction temperatures of Hi3516CV610

Chip	Junction Temperature for Working (T <sub>J</sub> )		Destructive Junction Temperature (°C)
	Min (°C)	Max (°C)	
Hi3516CV610	-20	105	125

**⚠ CAUTION**

- ☞ Under no condition can the chip junction temperature exceed the destructive junction temperature in [Table 2-15](#). If the chip junction temperature exceeds the destructive junction temperature, the chip may be physically damaged.
- ☞ In normal working conditions, the junction temperature of the chip must be within the working junction temperature range, and the ambient temperature of the chip must be within the working ambient temperature range.

[Table 2-15](#) lists the thermal resistance parameters.

**Table 2-15** Thermal resistance parameters

Parameter	Symbol	Typ		Unit
		Hi3516CV610-10B/20B/20S/20G	Hi3516CV610-00B/00S/00G	
Junction-to-ambient thermal resistance	θ <sub>JA</sub>	26.9	23.46	°C/W
Junction-to-board thermal resistance	θ <sub>JB</sub>	10.3	10.60	°C/W
Junction-to-case thermal resistance	θ <sub>JC</sub>	7.4	5.34	°C/W

## 2.5.3 Operating Conditions

[Table 2-16](#) describes the operating condition of the core power supply when the SVB solution is used.



**Table 2-16** Operating condition of the core power supply

Chip	Symbol	Description	SVB Voltage			Unit
			Min	Max	Error Tolerance	
Hi3516CV610 -10B	VDD	Core power	0.843	0.950	: 0.03	V
Hi3516CV610 -20B/20S/20G	VDD	Core power	0.838	0.990	: 0.03	V
Hi3516CV610 -00B/00S/00G	VDD	Core power	0.838	0.990	: 0.03	V

**NOTICE**

The confirmed result of the SVB\_CHECK tool should be used as the standard for the configured value of the power voltage.

[Table 2-17](#) and [Table 2-18](#) describe the operating conditions of other power supplies.

**Table 2-17** Operating conditions for the power supplies of Hi3516CV610-10B/20B/20S/20G under normal voltage

Symbol	Description	Min	Typ	Max	Unit
DVDD33	I2C/I2S/PWM/UART/GPIO/JTAG IO power	2.97	3.3	3.63	V
DVDD3318_FLASH	SDIO/SFC/EMMC power	2.97/1.71	3.3/1.8	3.63/1.89	V
DVDD3318_Sensor	Sensor IO power	2.97/1.71	3.3/1.8	3.63/1.89	V
AVDD18_MIPIRX	1.8 V MIPIRX analog power	1.71	1.8	1.89	V
	1.8 V CMOS mode IO power	1.71	1.8	1.89	V
AVDD33_AC_U2	3.3 V AUDIO/U2 analog power	2.97	3.3	3.63	V
AVDD_BAT	RTC battery power	1.6	3.0	3.6	V



Symbol	Description	Min	Typ	Max	Unit
AVDD33_FE	3.3 V FEPHY analog power	3.145	3.3	3.465	V
AVDD33_PLL	3.3 V PLL analog power	2.97	3.3	3.63	V
AVDD33_DDRPLL	3.3 V DDRPLL analog power	2.97	3.3	3.63	V
VDDIO_DDR	DDR2 IO power	1.71/1.425	1.8/1.5	1.89/1.575	V
	DDR3/DDR3L IO power	1.425/1.282	1.5/1.35	1.575/1.417	V

**Table 2-18** Operating conditions for the power supplies of Hi3516CV610-00B/00S/00G under normal voltage

Symbol	Description	Min	Typ	Max	Unit
DVDD33	I2C/I2S/PWM/UART/GPIO/JTAG IO power	2.97	3.3	3.63	V
DVDD3318_FLASH	SDIO/SFC/EMMC power	2.97/1.71	3.3/1.8	3.63/1.89	V
DVDD3318	SPI0/SENROR_HS/V_S power	2.97	3.3	3.63	V
DVDD3318_Sensor	Sensor IO power	2.97/1.71	3.3/1.8	3.63/1.89	V
AVDD33_U2	3.3 V USB2 analog power	2.97	3.3	3.63	V
AVDD18_MIPIRX	1.8 V MIPIRX analog power	1.71	1.8	1.89	V
	1.8 V CMOS mode IO power	1.71	1.8	1.89	V
AVDD33_AC	3.3 V AUDIO analog power	2.97	3.3	3.63	V
AVDD_BAT	RTC battery power	1.6	3.0	3.6	V



Symbol	Description	Min	Typ	Max	Unit
AVDD33_FE	3.3 V FEPHY analog power	3.145	3.3	3.465	V
AVDD33_PLL	3.3 V PLL analog power	2.97	3.3	3.63	V
AVDD33_DDRPLL	3.3 V DDRPLL analog power	2.97	3.3	3.63	V
VDDIO_CK_DDR	DDR3 CLK IO power	1.425	1.5	1.575	V
VDDIO_DDR	DDR3 IO power	1.425	1.5	1.575	V

**Table 2-19** Destructive voltage values of Hi3516CV610-10B/20B/20S/20G

Parameter Description	Power-Supply Pins	Min (V)	Max (V)
Core power supply voltage	VDD	-0.2	1.17
DDR I/O power supply voltage	VDDIO_DDR	-0.2	1.8/2.16
3.3 V or 1.8 V I/O power supply voltage	DVDD3318_FLASH/DVDD3318_SENSOR/AVDD18_MIPIRX	-0.2	3.96/2.16
3.3 V IO/IP power supply voltage	AVDD33_AC_U2/AVDD33_DDR_PLL/DVDD33/AVDD33_PLL/AVDD33_FE	-0.2	3.96
RTC battery power	AVDD_BAT	-0.2	3.96

**Table 2-20** Destructive voltage values of Hi3516CV610-00B/00S/00G

Parameter Description	Power-Supply Pins	Min (V)	Max (V)
Core power supply voltage	VDD	-0.2	1.17
DDR I/O power supply voltage	VDDIO_DDR/VDDIO_DDR_CK	-0.2	1.8



Parameter Description	Power-Supply Pins	Min (V)	Max (V)
3.3 V or 1.8 V I/O power supply voltage	DVDD3318_FLASHDVD D3318_SENSOR/AVDD1 8_MIPIRX/DVDD3318	-0.2	3.96/2.16
3.3 V IO/IP power supply voltage	AVDD33_AC/AVDD33_D DR_PLL/DVDD33/AVDD 33_PLL/AVDD33_FE/ AVDD33_U2	-0.2	3.96
RTC battery power	AVDD_BAT	-0.2	3.96

#### CAUTION

- As long as the power supply voltage of the chip exceeds the destructive voltage range, the physical functions of the chip may be permanently damaged.
- The power supply voltage of the chip must be within the range of the working conditions of the power supply. Otherwise, the quality cannot be ensured.

## 2.5.4 Power-On and Power-Off Sequences

For details about power-on and power-off sequences, see section 1.2.5 "Requirements on the Power-on and Power-off Timing" in the *Hi35xxVxxx Hardware Design User Guide*.

## 2.5.5 DC and AC Electrical Parameters

#### CAUTION

- If the LED needs to be driven, pay attention to the restrictions on the DC current.
- If a pull-up or pull-down resistor needs to be connected, ensure that the resistance is greater than or equal to 4.7 kilohms.

[Table 2-21](#) and [Table 2-22](#) describe the direct current (DC) electrical parameters.

**Table 2-21** DC electrical parameters

(DVDD33/DVDD3318\_Sensor/DVDD\_3318(BGA)/DVDD3318\_FLASH=3.3V)

Symbol	Description	Min	Typ	Max	Unit	Remarks
DVDD33	Interface voltage	2.97	3.3	3.63	V	-



Symbol	Description	Min	Typ	Max	Unit	Remarks
$V_{IH}$	High-level input voltage	2.1	-	-	V	Only the USB2_VBUS interface is compatible with the 5 V input.
$V_{IL}$	Low-level input voltage	-	-	0.7	V	-
$I_L$	Input leakage current	-	-	$\pm 10$	$\mu A$	-
$I_{Oz}$	Tristate output leakage current	-	-	$\pm 10$	$\mu A$	-
$V_{OH}$	High-level output voltage	2.4	-	-	V	-
$V_{OL}$	Low-level output voltage	-	-	0.4	V	-
$I_{OH}$	High-level output current	4.543	-	-	mA	IO2_level 1
		8.952	-	-	mA	IO2_level 2
		13.34	-	-	mA	IO2_level 3
		17.73	-	-	mA	IO2_level 4
$I_{OL}$	Low-level output current	3.03	-	-	mA	IO2_level 1
		6.018	-	-	mA	IO2_level 2
		8.985	-	-	mA	IO2_level 3
		11.88	-	-	mA	IO2_level 4
$I_{OH}$	High-level output current	3.647	-	-	mA	IO6_level 1
		7.284	-	-	mA	IO6_level 2
		14.26	-	-	mA	IO6_level 3
		17.89	-	-	mA	IO6_level 4
		28.38	-	-	mA	IO6_level 5
		32	-	-	mA	IO6_level 6
		38.95	-	-	mA	IO6_level 7
		42.57	-	-	mA	IO6_level 8
$I_{OL}$	Low-level output current	2.428	-	-	mA	IO6_level 1
		4.849	-	-	mA	IO6_level 2



Symbol	Description	Min	Typ	Max	Unit	Remarks
I <sub>OH</sub>	High-level output current	9.604	-	-	mA	IO6_level 3
		11.93	-	-	mA	IO6_level 4
		18.94	-	-	mA	IO6_level 5
		21.32	-	-	mA	IO6_level 6
		25.99	-	-	mA	IO6_level 7
		28.27	-	-	mA	IO6_level 8
I <sub>OL</sub>	Low-level output current	4.557	-	-	mA	EMMC IO_level 1
		9.112	-	-	mA	EMMC IO_level 2
		13.41	-	-	mA	EMMC IO_level 3
		17.96	-	-	mA	EMMC IO_level 4
		26.65	-	-	mA	EMMC IO_level 5
		31.19	-	-	mA	EMMC IO_level 6
		35.48	-	-	mA	EMMC IO_level 7
		40.01	-	-	mA	EMMC IO_level 8
		44.29	-	-	mA	EMMC IO_level 9
		48.84	-	-	mA	EMMC IO_level 10
		53.13	-	-	mA	EMMC IO_level 11
		57.67	-	-	mA	EMMC IO_level 12
		66.33	-	-	mA	EMMC IO_level 13
		70.87	-	-	mA	EMMC IO_level 14
		75.14	-	-	mA	EMMC IO_level 15
		79.67	-	-	mA	EMMC IO_level 16



Symbol	Description	Min	Typ	Max	Unit	Remarks
		23.78	-	-	mA	EMMC IO_level 7
		26.73	-	-	mA	EMMC IO_level 8
		29.47	-	-	mA	EMMC IO_level 9
		32.44	-	-	mA	EMMC IO_level 10
		35.45	-	-	mA	EMMC IO_level 11
		38.41	-	-	mA	EMMC IO_level 12
		44.09	-	-	mA	EMMC IO_level 13
		47.04	-	-	mA	EMMC IO_level 14
		49.9	-	-	mA	EMMC IO_level 15
		52.84	-	-	mA	EMMC IO_level 16
R <sub>PU1</sub>	Internal pull-up resistance	33.35	42.43	53.51	kΩ	-
R <sub>PD1</sub>	Internal pull-down resistance	28.62	37.48	48.45	kΩ	-
R <sub>PU2</sub>	Internal pull-up resistance	6.451	7.761	9.119	kΩ	-
R <sub>pu4</sub>	Internal pull-up resistance	18.96	25.82	35.21	kΩ	-
R <sub>PD4</sub>	Internal pull-down resistance	51.29	92.86	152	kΩ	-
R <sub>PD5</sub>	Internal pull-down resistance	17.95	27.14	40.16	kΩ	-
R <sub>PU7</sub>	Internal pull-up resistance	18.14	24.69	33.59	kΩ	-
R <sub>PD7</sub>	Internal pull-down resistance	51.38	92.48	150.3	kΩ	-
R <sub>PU11</sub>	Internal pull-up resistance	18.36	25	34.04	kΩ	-
R <sub>PD11</sub>	Internal pull-down resistance	18.25	27.65	50.96	kΩ	-
R <sub>PU12</sub>	Internal pull-up resistance	6.413	7.796	9.239	kΩ	-



Symbol	Description	Min	Typ	Max	Unit	Remarks
R <sub>PU13</sub>	Internal pull-up resistance	18.42	25.08	34.15	kΩ	-
R <sub>PD13</sub>	Internal pull-down resistance	18.17	27.56	40.95	kΩ	-
R <sub>PU20</sub>	Internal pull-up resistance	24.39	29.48	35.89	kΩ	-
R <sub>PD20</sub>	Internal pull-down resistance	22.78	28.2	34.14	kΩ	-
R <sub>PU21</sub>	Internal pull-up resistance	24.39	29.48	35.89	kΩ	-
R <sub>PD21</sub>	Internal pull-down resistance	22.78	28.2	34.14	kΩ	-

**Table 2-22** DC electrical parameters (DVDD3318\_FLASH/DVDD3318\_Sensor/DVDD3318(BGA)=1.8V)

Symbol	Description	Min	Typ	Max	Unit	Remarks
DVDD18	Interface voltage	1.62	1.8	1.98	V	-
V <sub>IH</sub>	High-level input voltage	0.65×DVDD18	-	-	V	Only the USB2_VBUS interface is compatible with the 5 V input.
V <sub>IL</sub>	Low-level input voltage	-	-	0.4	V	-
I <sub>L</sub>	Input leakage current	-	-	±10	μA	-
I <sub>OZ</sub>	Tristate output leakage current	-	-	±10	μA	-
V <sub>OH</sub>	High-level output voltage	1.4	-	-	V	-
V <sub>OL</sub>	Low-level output voltage	-	-	0.45	V	-
I <sub>OH</sub>	High-level output current	1.006	-	-	mA	IO2_level 1
		1.974	-	-	mA	IO2_level 2
		2.94	-	-	mA	IO2_level 3



Symbol	Description	Min	Typ	Max	Unit	Remarks
		3.905	-	-	mA	IO2_level 4
I <sub>OL</sub>	Low-level output current	1.431	-	-	mA	IO2_level 1
		2.854	-	-	mA	IO2_level 2
		4.282	-	-	mA	IO2_level 3
		5.696	-	-	mA	IO2_level 4
I <sub>OH</sub>	High-level output current	0.8064	-	-	mA	IO6_level 1
		1.612	-	-	mA	IO6_level 2
		3.136	-	-	mA	IO6_level 3
		3.941	-	-	mA	IO6_level 4
		6.233	-	-	mA	IO6_level 5
		7.037	-	-	mA	IO6_level 6
		8.556	-	-	mA	IO6_level 7
		9.359	-	-	mA	IO6_level 8
I <sub>OL</sub>	Low-level output current	1.144	-	-	mA	IO6_level 1
		2.289	-	-	mA	IO6_level 2
		4.568	-	-	mA	IO6_level 3
		5.701	-	-	mA	IO6_level 4
		9.106	-	-	mA	IO6_level 5
		10.25	-	-	mA	IO6_level 6
		12.52	-	-	mA	IO6_level 7
		13.64	-	-	mA	IO6_level 8
I <sub>OH</sub>	High-level output current	1.008	-	-	mA	EMMC IO_level 1
		2.017	-	-	mA	EMMC IO_level 2
		2.95	-	-	mA	EMMC IO_level 3
		3.957	-	-	mA	EMMC IO_level 4
		5.885	-	-	mA	EMMC IO_level 5
		6.861	-	-	mA	EMMC IO_level 6
		7.793	-	-	mA	EMMC IO_level 7



Symbol	Description	Min	Typ	Max	Unit	Remarks
I <sub>OL</sub>	Low-level output current	8.798	-	-	mA	EMMC IO_level 8
		9.729	-	-	mA	EMMC IO_level 9
		10.74	-	-	mA	EMMC IO_level 10
		11.67	-	-	mA	EMMC IO_level 11
		12.68	-	-	mA	EMMC IO_level 12
		14.57	-	-	mA	EMMC IO_level 13
		15.57	-	-	mA	EMMC IO_level 14
		16.51	-	-	mA	EMMC IO_level 15
		17.51	-	-	mA	EMMC IO_level 16
		1.431	-	-	mA	EMMC IO_level 1
		2.854	-	-	mA	EMMC IO_level 2
		4.292	-	-	mA	EMMC IO_level 3
		5.713	-	-	mA	EMMC IO_level 4
		8.557	-	-	mA	EMMC IO_level 5
		9.977	-	-	mA	EMMC IO_level 6
		11.4	-	-	mA	EMMC IO_level 7
		12.82	-	-	mA	EMMC IO_level 8
		14.22	-	-	mA	EMMC IO_level 9
		15.64	-	-	mA	EMMC IO_level 10
		17.07	-	-	mA	EMMC IO_level 11
		18.49	-	-	mA	EMMC IO_level 12
		21.31	-	-	mA	EMMC IO_level 13
		22.73	-	-	mA	EMMC IO_level 14
		24.15	-	-	mA	EMMC IO_level 15
		25.56	-	-	mA	EMMC IO_level 16
R <sub>pu2</sub>	Internal pull-up resistance	6.519	7.957	9.504	kΩ	-
R <sub>pu4</sub>	Internal pull-up resistance	33.75	53.05	81.19	kΩ	-



Symbol	Description	Min	Typ	Max	Unit	Remarks
R <sub>pd4</sub>	Internal pull-down resistance	109.3	208.5	373.7	kΩ	-
R <sub>pd5</sub>	Internal pull-down resistance	31.32	55.16	93.83	kΩ	-
R <sub>PU12</sub>	Internal pull-up resistance	6.48	7.99	9.618	kΩ	-
R <sub>pu13</sub>	Internal pull-up resistance	32.05	50.16	76.21	kΩ	-
R <sub>pd13</sub>	Internal pull-down resistance	32.03	55.97	95.71	kΩ	-
R <sub>PU20</sub>	Internal pull-up resistance	26.28	33.23	43.47	kΩ	-
R <sub>PD20</sub>	Internal pull-down resistance	31.15	23.98	40.66	kΩ	-
R <sub>PU21</sub>	Internal pull-up resistance	26.28	33.23	43.47	kΩ	-
R <sub>PD21</sub>	Internal pull-down resistance	31.15	23.98	40.66	kΩ	-

## 2.5.6 MIPI/LVDS Rx Electrical Parameters

Table 2-23 describes the differential DC electrical parameters of DPHY, HiSPi, Sub-LVDS and low-voltage differential signaling (LVDS).

Table 2-23 Differential DC electrical parameters

Symbol	Description		Min	Typ	Max	Unit
VIDTH(SL)	Differential Input Threshold Voltage  (VP-VM)	Sub-LVDS	70	-	-	mV
VIDTH(HS)		HiSPi(SLVS)	70	-	-	
VIDTH(HiVCM)		HiSPi(HiVC M)	100	-	-	
VIDTH(DP)		D-PHY HS	70	-	-	
VIDTH(LV)		LVDS	100	-	-	
VIDTH(ML)		Mini-LVDS	100	-	-	



Symbol	Description	Min	Typ	Max	Unit
VCM(SL)	Common Mode Voltage Range (VP+VM)/2	Sub-LVDS	0.5	0.9	1.3
VCM(HS)		HiSPi(SLVS)	0.07	0.2	0.35
VCM(HiVCM)		HiSPi(HiVCM)	0.66	0.90	1.17
VCM(DP)		D-PHY HS	0.07	0.2	0.33
VCM(LV)		LVDS	0.925	1.2	1.475
VCM(ML)		Mini-LVDS	1.025	1.2	1.375
VISVR (SL)	Single-ended Input Voltage Range VP, VM	Sub-LVDS	0.4	-	1.4
VCM(HS)		HiSPi	-0.165	-	0.575
VCM(DP)		D-PHY HS	-0.04	-	0.46
VCM(LV)		LVDS	0	-	1.8
VCM(ML)		Mini-LVDS	0.825	-	1.575
ZID(SL)	Internal Termination Resistor Value	Sub-LVDS	80	100	125 ohm
ZID(HS)		HiSPi			
ZID(LV)		LVDS			
ZID(ML)		Mini-LVDS			
ZID(DP)		D-PHY HS			

[Table 2-24](#) and [Table 2-25](#) describe the MIPI DPHY parameters.

**Table 2-24** MIPI DPHY high-speed (HS) DC parameters

Symbol	Description	Min	Typ	Max	Unit
VTERM-EN	Single-ended threshold for HS termination enable	-	-	450	mV

**Table 2-25** MIPI DPHY high-speed (HS) AC parameters

Symbol	Description	Min	Typ	Max	Unit
$\Delta VCMRX(HF)$	Common-mode interface beyond 450 MHz	-	-	100	-



Symbol	Description	Min	Typ	Max	Unit
$\Delta VCMRX(LF)$	Common-mode interface 50 MHz–450 MHz	-50	-	50	mV
CCM	Common-mode termination	-	-	60	pF

**Table 2-26** MIPI DPHY low-power (LP) DC parameters

Symbol	Description	Min	Typ	Max	Unit
VIHLP	Logic 1 input voltage	880	-	-	mV
VILLP	Logic 0 input voltage	-	-	550	
VHYST	Input hysteresis	25	-	-	

**Table 2-27** MIPI DPHY low-power (LP) AC parameters

Symbol	Description	Min	Typ	Max	Unit
$e_{SPIKE}$	Input pulse rejection	-	-	300	V·ps
$T_{MIN-RX}$	Minimum pulse rejection	20	-	-	ns
$V_{INT}$	Peak interference amplitude	-	-	200	mV
$f_{INT}$	Interference frequency	450	-	-	MHz

## 2.5.7 Audio CODEC Electrical Parameters

Table 2-28 to Table 2-32 describe the electrical parameters of the audio CODEC.

**Table 2-28** Overall specifications

Description	Min	Typ	Max	Unit	Remarks
Analog circuit power AVDD33_U2_AC	3	3.3	3.6	V	Relative to AVSS_AC
AC_VREF	-	AVDD33_U2_AC /2	-	V	Relative to AVSS_AC



**Table 2-29** Major DAC specifications

Description	Min	Typ	Max	Unit	Remarks
Output amplitude at full scale	-	1	-	Vrms	Maximum output signal swing

**Table 2-30** Major ADC specifications

Description	Min	Typ	Max	Unit	Remarks
Maximum input amplitude	-	1	2	Vrms	Maximum input signal swing of the ADC (maximum single-ended voltage: 1 V; differential voltage: 2 V)

**Table 2-31** Major MICBIAS specifications

Description	Min	Typ	Max	Unit	Remarks
Bias voltage	-	$2.1 \times AVDD3_3 / U2\_AC / 3.3$	$2.8 \times AVDD33\_U2\_AC / 3.3$	V	MIC bias voltage
Maximum output current	-	-	3	mA	-

**Table 2-32** Major MICPGA specifications

Description	Min	Typ	Max	Unit	Remarks
Input voltage range	-	1	2	Vrms	Maximum input signal swing (maximum single-ended voltage: 1 V; differential voltage: 2 V)
Input impedance	-	30	66	kilohm	MICPGA input impedance (single-ended input: 30 kilohms; differential input: 60 kilohms)

## 2.6 Interface Timings

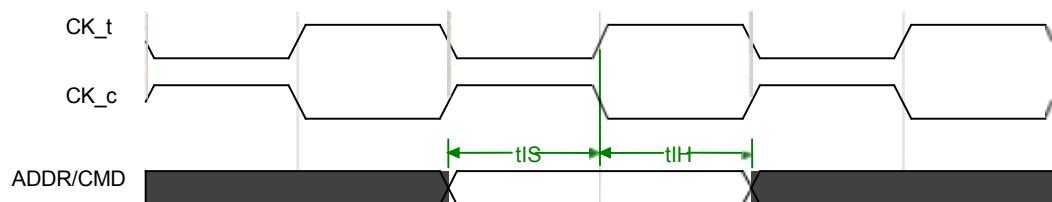
### 2.6.1 DDR Interface Timings

#### 2.6.1.1 Write Timings

##### Write Timings of CMD/ADDR Relative to CK

For the write timing of DDR CMD/ADDR relative to CK, tIS is the setup time from CMD/ADDR to the CK rising edge, and tIH is the hold time from the CK rising edge to CMD/ADDR. Figure 2-12 shows the write timing of CMD/ADDR relative to CK for the DDR3/3L SDRAM.

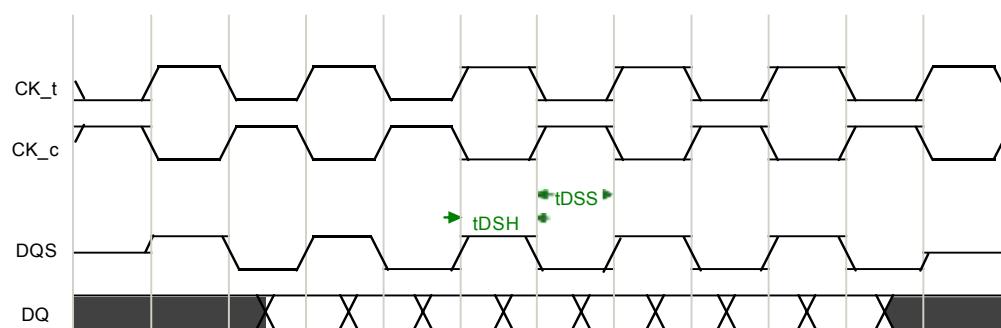
**Figure 2-12** Write timing of CMD/ADDR relative to CK for the DDR3/3L SDRAM



##### Write Timings of DQS Relative to CK

For the write timing of DQS relative to CK for the DDR3/3L SDRAM, tDSH is the hold time from the CK rising edge to the DQS falling edge, and tDSS is the setup time from the DQS falling edge to the CK rising edge. Figure 2-13 shows the write timing of DQS relative to CK.

**Figure 2-13** Write timing of DQS relative to CK for the DDR3/3L SDRAM



### 2.6.1.2 Read Timings

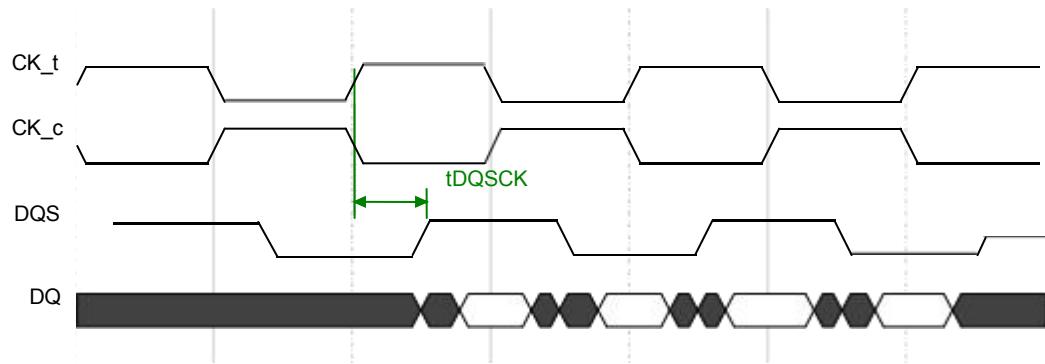
#### Read Timings of CMD/ADDR Relative to CK

The read timings of CMD/ADDR relative to CK are the same as the "[Write Timings of CMD/ADDR Relative to CK](#)."

#### Read Timings of CK Relative to DQS

For the read timings of CK relative to DQS for the DDR3/3L SDRAM, tDQSCCK is the skew of the valid DQS relative to CK. [Figure 2-14](#) shows the read timings of CK relative to DQS for the DDR3/3L SDRAM.

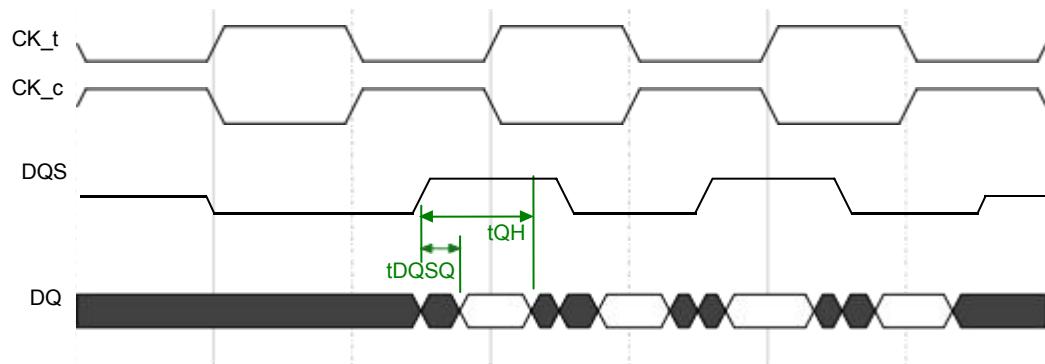
**Figure 2-14** Read timings of CK relative to DQS for the DDR3/3L SDRAM



#### Read Timings of DQS Relative to DQ

For the read timings of DQS relative to DQ for the DDR3/3L SDRAM, tQH is the earliest invalid DQ jitter relative to DQS. [Figure 2-15](#) shows the read timings of DQS relative to DQ for the DDR3/3L SDRAM.

**Figure 2-15** Read timings of DQS relative to DQ for the DDR3/3L SDRAM





### 2.6.1.3 DDR Timing Parameters

The timings of the DDR interface comply with the JEDEC (JESD79-3F) standards. All the timings in this document are output on the DDR PHY side.

[Table 2-33](#)describes the CK parameters of the DDR3 SDRAM (DDR3/3L-2133).

**Table 2-33** CK parameters of the DDR3 SDRAM (DDR3/3L-2133)

Parameter	Min	Typ	Max	Unit
DDR clock frequency	-	1066.5	-	MHz
PLL jitter tJIT (cc)	-100	-	100	ps
PLL jitter tJIT (per)	-50	-	50	ps
PLL duty ratio (avg)	47.000		53.000	%

[Table 2-34](#)describes the DQS parameters of the DDR3 SDRAM (DDR3/3L-2133).

**Table 2-34** DQS parameters of the DDR3 SDRAM (DDR3/3L-2133)

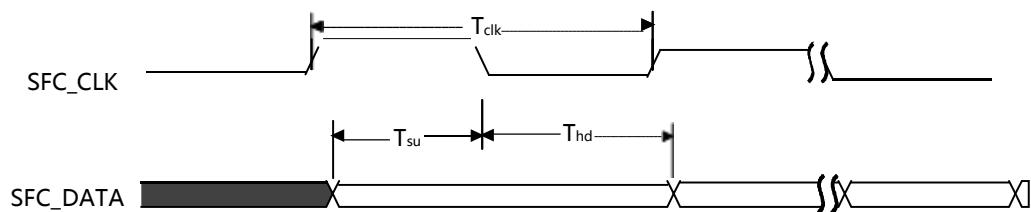
Parameter	Symbol	Min	Max	Unit
Setup time of the DQS falling edge relative to the DDR clock	tDSS	0.18	-	tCK
Hold time of the DQS falling edge relative to the DDR clock	tDSH	0.18	-	tCK
Skew of DQS relative to DQ	tDQSQ	-	75	ps
ADDR/CMD setup time relative to the DDR clock referenced to $V_{IH(AC)}$ and $V_{IL(AC)}$ levels	tIS(base)	135	-	ps
ADDR/CMD hold time relative to the DDR clock referenced to $V_{IH(DC)}$ and $V_{IL(DC)}$ levels	tIH(base)	100	-	-
Skew of the DQS output relative to the DDR clock	tDQSCK	-180	180	ps

### 2.6.2 SPI FLASH Interface Timings

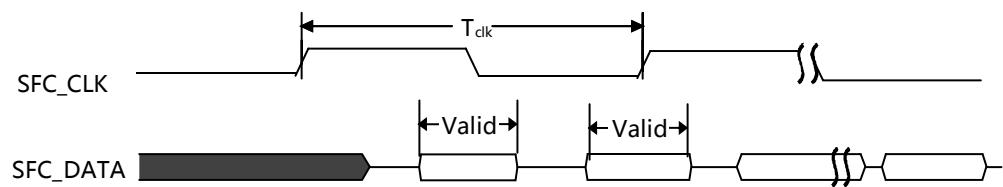
[Figure 2-16](#)and[Figure 2-17](#)show the SPI FLASH input timings.



**Figure 2-16 SPI FLASH input timing (SDR mode)**



**Figure 2-17 SPI FLASH input timing (DDR mode)**



[Table 2-35](#) and [Table 2-36](#) describe the SPI FLASH input timing parameters.

**Table 2-35** SPI flash input timing parameters (voltage = 3.3 V)

Parameter	Symbol	Min	Typ	Max	Unit
Clock cycle of SPI FLASH_CLK (SDR mode)	$T_{clk}$	10.1	-	83.2	ns
Clock cycle of SPI FLASH_CLK (DDR mode)	$T_{clk}$	10.1	-	83.2	ns
Clock duty cycle of SFC_CLK (DDR mode)	Duty	45	-	55	%
Input signal setup time (SDR mode)	$T_{su}$	2	-	-	ns
Input signal hold time (SDR mode)	$T_{hd}$	1	-	-	ns
Valid time of the input signal (DDR)	Valid	3	-	-	ns

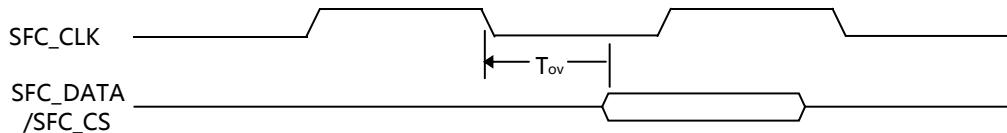


**Table 2-36** SPI flash input timing parameters (voltage = 1.8 V)

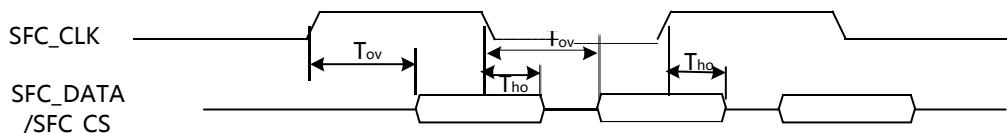
Parameter	Symbol	Min	Typ	Max	Unit
Clock cycle of SPI FLASH_CLK (SDR mode)	$T_{clk}$	13.33	-	83.2	ns
Clock cycle of SPI FLASH_CLK (DDR mode)	$T_{clk}$	13.33	-	83.2	ns
Clock duty cycle of SFC_CLK (DDR mode)	Duty	45	-	55	%
Input signal setup time (SDR mode)	$T_{su}$	1.9	-	-	ns
Input signal hold time (SDR mode)	$T_{hd}$	0	-	-	ns
Valid time of the input signal (DDR)	Valid	3.2	-	-	ns

[Figure 2-18](#) and [Figure 2-19](#) show the SPI FLASH output timings.

**Figure 2-18** SPI FLASH output timing (SDR mode)



**Figure 2-19** SPI FLASH output timing (DDR mode)



[Table 2-37](#) and [Table 2-38](#) describe the SPI FLASH output timing parameters.

**Table 2-37** SPI flash output timing parameters (voltage = 3.3 V)

Parameter	Symbol	Min	Typ	Max	Unit
Clock cycle of SPI SFCLK (SDR mode)	T	10.1	-	83.2	ns



Parameter	Symbol	Min	Typ	Max	Unit
Clock cycle of SPI SFCLK (DDR mode)	T	10.1	-	83.2	ns
Output data signal delay (SDR mode)	$T_{ov}$	-1	-	3	ns
Output data signal delay (DDR mode)	$T_{ov}$	-	-	$T/2-1.5$	ns
Hold time of the output signal (DDR)	$T_{ho}$	1.5	-	-	ns

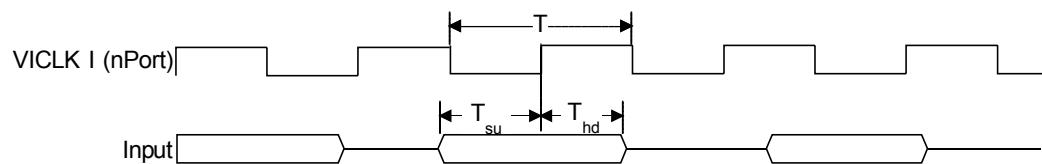
**Table 2-38** SPI flash output timing parameters (voltage = 1.8 V)

Parameter	Symbol	Min	Typ	Max	Unit
Clock cycle of SPI SFCLK (SDR mode)	T	13.33	-	83.2	ns
Clock cycle of SPI SFCLK (DDR mode)	T	13.33	-	83.2	ns
Output data signal delay (SDR mode)	$T_{ov}$	-1	-	3	ns
Output data signal delay (DDR mode)	$T_{ov}$	-	-	$T/2-1.5$	ns
Hold time of the output signal (DDR)	$T_{ho}$	1.5	-	-	ns

## 2.6.3 VI Interface Timings

Figure 2-20 shows the VI interface timing in CMOS mode.

**Figure 2-20** VI interface timings in CMOS mode



**Table 2-39** describes the VI interface timing parameters.

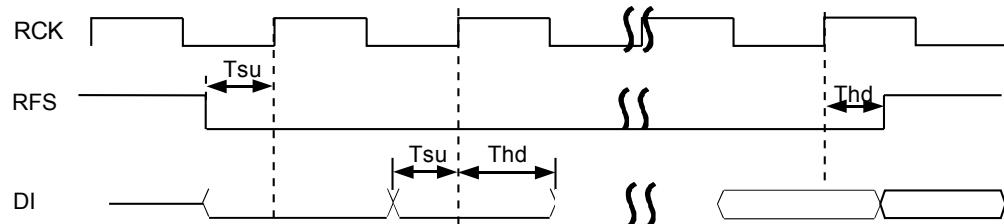
**Table 2-39** VI interface timing parameters

Parameter	Symbol	Min	Typ	Max	Unit
VICLK clock cycle	T	6.73	-	-	ns
Input signal setup time	$T_{su}$	2.88	-	-	ns
Input signal hold time	$T_{hd}$	0.8	-	-	ns

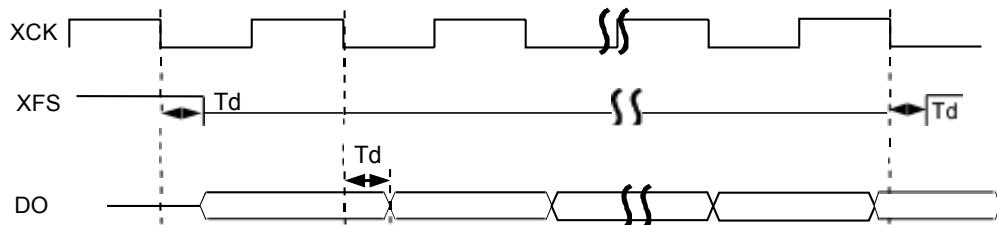
## 2.6.4 AIAO Interface Timings

### 2.1.1.2 I<sup>2</sup>S Interface Timings

[Figure 2-21](#) shows the RX timing of the I<sup>2</sup>S interface.

**Figure 2-21** RX timing of the I<sup>2</sup>S interface

[Figure 2-22](#) shows the TX timing of the I<sup>2</sup>S interface.

**Figure 2-22** TX timing of the I<sup>2</sup>S interface

[Table 2-40](#) describes the timing parameters of the I<sup>2</sup>S interface.

**Table 2-40** Timing parameters of the I<sup>2</sup>S interface

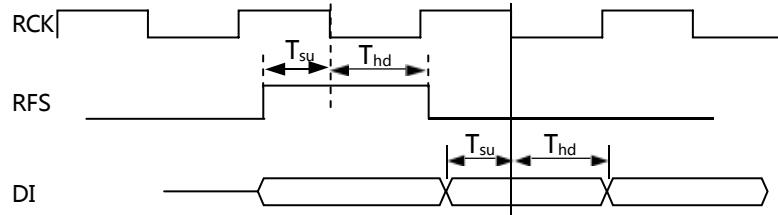
Parameter	Symbol	Min	Typ	Max	Unit
Input signal setup time	$T_{su}$	10	-	-	ns
Input signal hold time	$T_{hd}$	10	-	-	ns
Output signal delay	$T_d$	0	-	8	ns



### 2.1.1.3 PCM Interface Timings

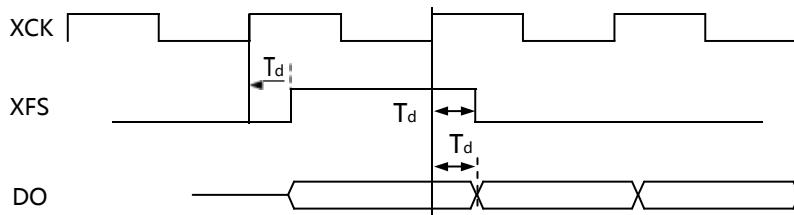
[Figure 2-23](#) shows the RX timing of the PCM interface.

**Figure 2-23** RX timing of the PCM interface



[Figure 2-24](#) shows the TX timing of the PCM interface.

**Figure 2-24** TX timing of the PCM interface



[Table 2-41](#) describes the timing parameters of the PCM interface.

**Table 2-41** Timing parameters of the PCM interface

Parameter	Symbol	Min	Typ	Max	Unit
Input signal setup time	$T_{su}$	10	-	-	ns
Input signal hold time	$T_{hd}$	10	-	-	ns
Output signal delay	$T_d$	0	-	8	ns

### 2.6.5 I<sup>2</sup>C Interface Timing

[Figure 2-25](#) shows the I<sup>2</sup>C transfer timing.



Figure 2-25 I<sup>2</sup>C transfer timing

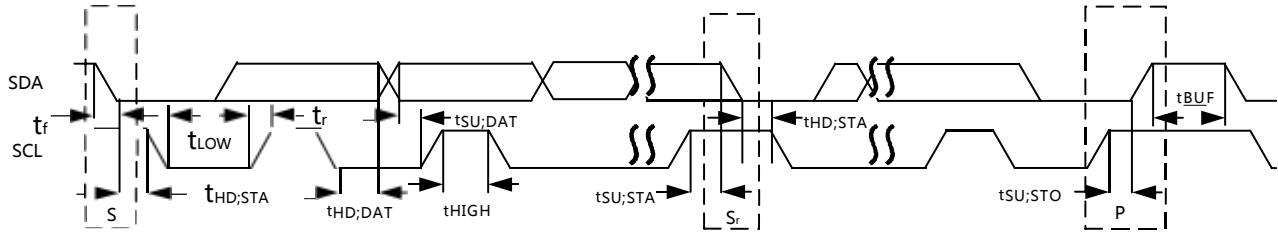


Table 2-42 describes the timing parameters of the I<sup>2</sup>C interface.

Table 2-42 Timing parameters of the I<sup>2</sup>C interface

Parameter	Symbol	Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	
Serial clock (SCL) frequency	f <sub>SCL</sub>	0	100	0	400	kHz
Start hold time	t <sub>HD;STA</sub>	4.0	-	0.6	-	μs
SCL low-level cycle	t <sub>LOW</sub>	4.7	-	1.3	-	μs
SCL high-level cycle	t <sub>HIGH</sub>	4.0	-	0.6	-	μs
Start setup time	t <sub>SU;STA</sub>	4.7	-	0.6	-	μs
Data hold time	t <sub>HD;DAT</sub>	0	3.45	0	0.9	μs
Data setup time	t <sub>SU;DAT</sub>	250	-	100	-	ns
Serial data (SDA) and SCL rising time	t <sub>r</sub>	-	1000	20+0.1C b	300	ns
SDA and SCL falling time	t <sub>f</sub>	-	300	20+0.1C b	300	ns
End setup time	t <sub>SU; STO</sub>	4.0	-	0.6	-	μs
Bus release time from start to end	t <sub>BUF</sub>	4.7	-	1.3	-	μs
Bus load	G <sub>b</sub>	-	400	-	400	pF

## 2.6.6 SPI Timings

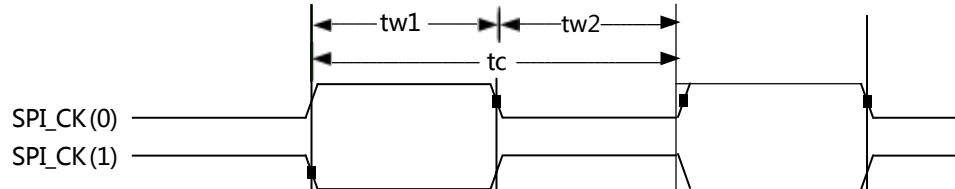
### D NOTE

In Figure 2-26 to Figure 2-28, the conventions are as follows:

- MSB = most significant bit
- LSB = least significant bit
- SPI\_CK(0): sph = 0
- SPI\_CK(1): sph = 1

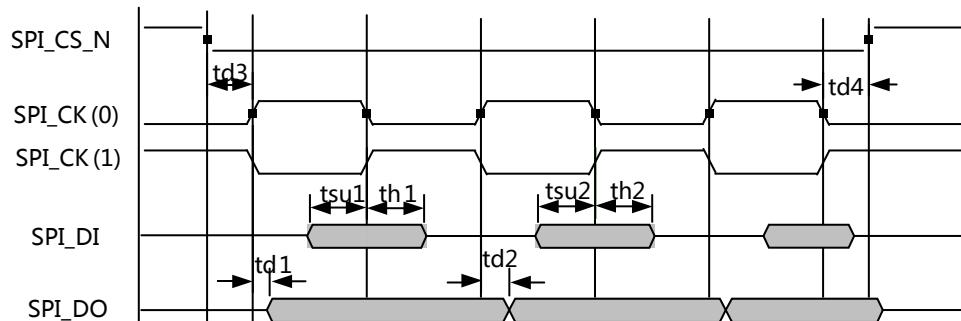
[Figure 2-26](#) shows the SPI clock (SPICK) timing.

**Figure 2-26** SPICK timing

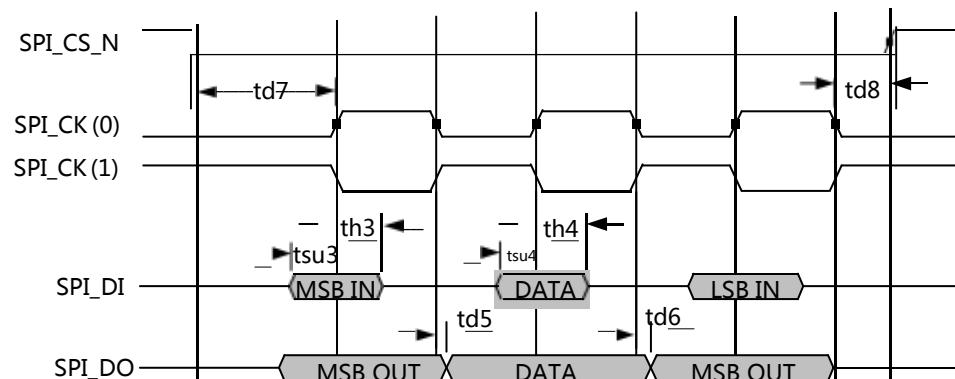


[Figure 2-27](#) and [Figure 2-28](#) show the SPI timings in master mode.

**Figure 2-27** SPI timing in master mode ( $sph = 1$ )



**Figure 2-28** SPI timing in master mode ( $sph = 0$ )



[Table 2-43](#) describes the SPI timing parameters.



**Table 2-43 SPI timing parameters**

Parameter	Symbol	Min	Typ	Max	Unit	Remark
Cycle time, SPI_CK	tc	spi0/1: 4000/Fsspc lk	-	spi0/1:4 000/Fss pclk	ns	Fsspclk unit: MHz For details about the Fsspclk frequency value, see chapter 12 "Peripherals."
Pulse duration, SPI_CK high (All Master Modes)	tw1	-	1/2 tc	-	ns	
Pulse duration, SPI_CK low (All Master Modes)	tw2		1/2 tc	-	ns	
Setup time, SPI_DI (input) valid before SPI_CK (output) falling edge	tsu1	spi0/1: 10	-	-	ns	
Setup time, SPI_DI (input) valid before SPICK (output) rising edge	tsu2	spi0/1: 10	-	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) falling edge	th1	spi0/1: 10	-	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) rising edge	th2	spi0/1: 10	-	-	ns	
Delay time, SPI_CK (output) rising edge to SPI_DO (output) transition	td1	spi0/1: -10	-	spi0/1: 10	ns	
Delay time, SPI_CK (output) falling edge to SPI_DO (output) transition	td2	spi0/1: -10	-	spi0/1: 10	ns	
Delay time, SPI_CS_N (output) falling edge to first SPI_CK (output) rising or falling edge	td3	spi0/1: (1/2 tc)-10	-	-	ns	



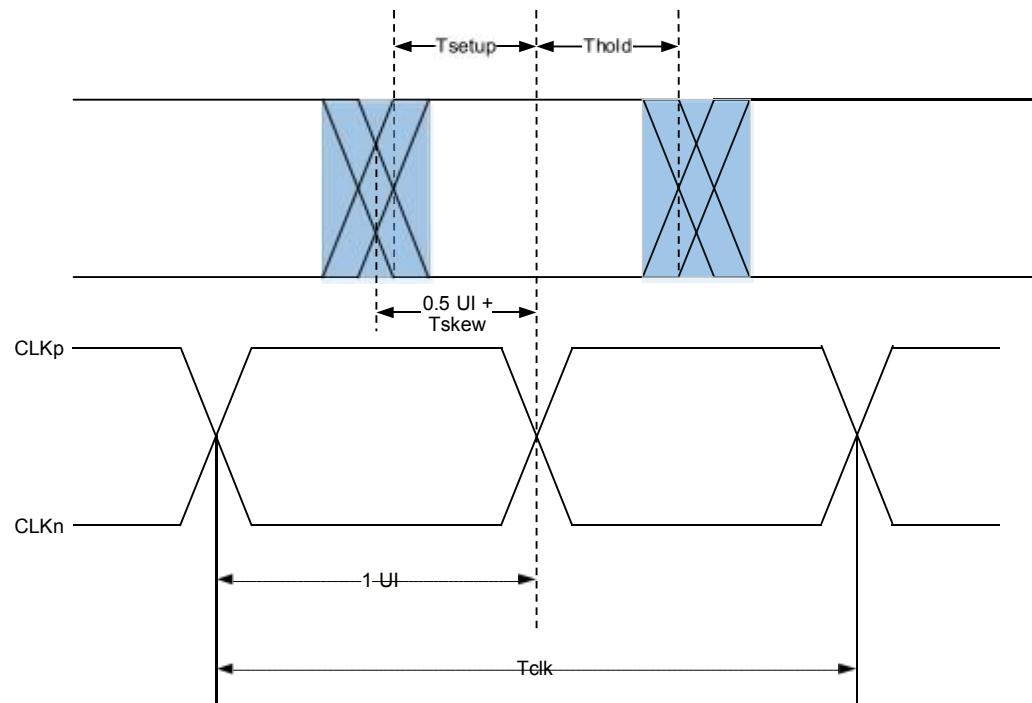
Parameter	Symbol	Min	Typ	Max	Unit	Remark
Delay time, SPI_CK (output) rising or falling edge to SPI_CS_N (output) rising edge	td4	spi0/1: tc-10	-	-	ns	
Setup time, SPI_DI (input) valid before SPI_CK (output) rising edge	tsu3	spi0/1: 10	-	-	ns	
Setup time, SPI_DI (in put) valid before SPI_CK (output) falling edge	tsu4	spi0/1: 10	-	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) rising edge	th3	spi0/1: 10	-	-	ns	
Hold time, SPI_DI (input) valid after SPI_CK (output) falling edge	th4	spi0/1: 10	-	-	ns	
Delay time, SPI_CK (output) falling edge to SPI_DO (output) transition	td5	spi0/1: -10	-	spi0/1: 10	ns	
Delay time, SPI_CK (output) rising edge to SPI_DO (output) transition	td6	spi0/1: -10	-	spi0/1: 10	ns	
Delay time, SPI_CS_N (output) falling edge to first SPI_CK (output) rising or falling edge	td7	spi0/1: tc-10	-	-	ns	
Delay time, SPI_CK (output) rising or falling edge to SPI_CS_N (output) rising edge	td8	spi0/1: (1/2 tc)-10	-	-	ns	

## 2.6.7 MIPI RX Timings

Figure 2-29 shows the clock timings of the MIPI RX DPHY/Sub-lvds/LVDS/HiSPi interface.



**Figure 2-29** Clock timings of the MIPI RX DPHY/Sub-lvds/LVDS/HiSPi interface



[Table 2-44](#)describes the MIPI RX DPHY/Sub-lvds/LVDS/HiSPi timing parameters.

**Table 2-44** MIPI RX DPHY/Sub-lvds/LVDS/HiSPi timing parameters

Symbol	Description	Min	Tye	Max	Unit
FMAX	Data rate	-	-	1.5G	bps
Tclk	Differential clock cycle	1.33	-	-	ns
Tskew	Data to clock skew ( $\leq 1$ Gbit/s)	$-0.15 \times UI$	-	$0.15 \times UI$	ns
	Data to clock skew ( $> 1$ Gbit/s)	$-0.2 \times UI$		$0.2 \times UI$	
Tsetup	Differential clock setup time ( $\leq 1$ Gbit/s)	$0.15 \times UI$	-	-	ns
	Differential clock setup time ( $> 1$ Gbit/s)	$0.2 \times UI$		-	
Thold	Differential clock hold time ( $\leq 1$ Gbit/s)	$0.15 \times UI$	-	-	ns



Symbol	Description	Min	Tye	Max	Unit
	Differential clock hold time (> 1 Gbit/s)	0.2×UI			

## D NOTE

UI is equal to  $T_{PERIOD}/2$ .

## 2.6.8 SDIO/MMC Interface Timings

### eMMC Interface Timings

Figure 2-30 shows the CMD/DATA input/output timings of the eMMC in DS/HS mode.

Figure 2-30 CMD/DATA input/output timings of the eMMC in DS/HS mode

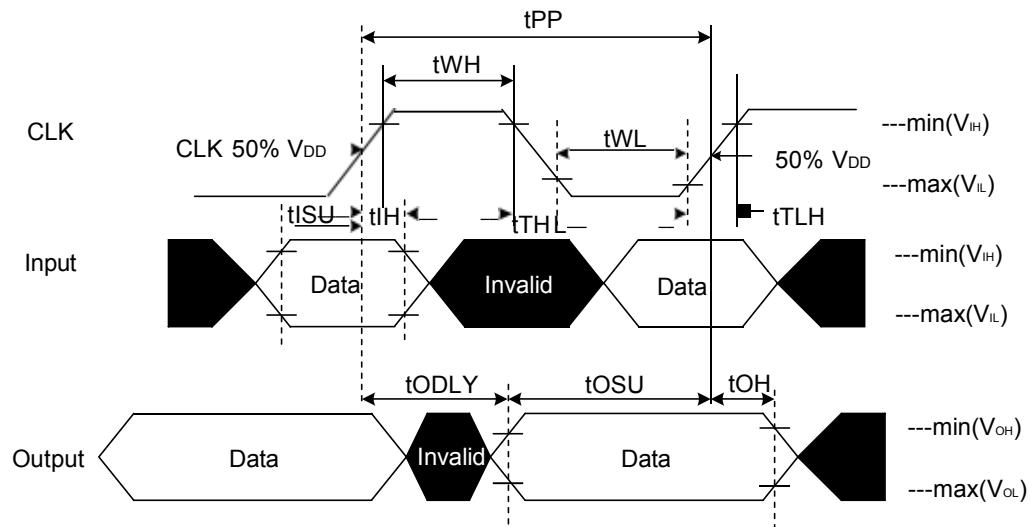


Table 2-45 describes the CMD/DATA input/output timing parameters of the eMMC in DS mode.

Table 2-45 CMD/DATA input/output timing parameters of the eMMC in DS mode

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
Clock CLK						
tpp	Cycle time data transfer mode	38.5	40.4	-	ns	-
tWH	Clock high time	10	18	-	ns	-



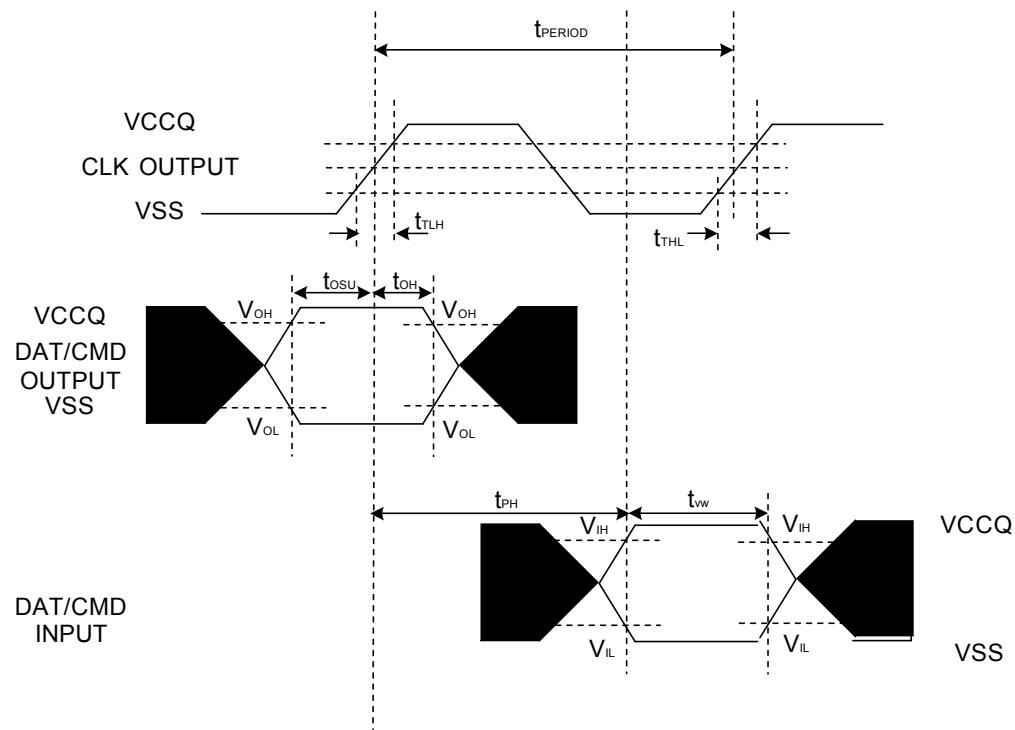
Symbol	Parameter	Min	Typ	Max	Unit	Remarks
tWL	Clock low time	10	18	-	ns	-
<b>Outputs DAT (referenced to CLK)</b>						
tosu	Output set-up time	5	18	-	ns	-
toH	Output hold time	5	18	-	ns	-
<b>Inputs DAT (referenced to CLK)</b>						
tISU	Input set-up time	11.7	-	-	ns	-
tIH	Input hold time	8.3	-	-	ns	-

**Table 2-46** describes the CMD/DATA input/output timing parameters of the eMMC in HS mode.

**Table 2-46** CMD/DATA input/output timing parameters of the eMMC in HS mode

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
<b>Clock CLK</b>						
tpp	Cycle time data transfer mode	19.2	20.2	-	ns	-
tWH	Clock high time	6.5	9	-	ns	-
tWL	Clock low time	6.5	8	-	ns	-
<b>Outputs DAT (referenced to CLK)</b>						
tosu	Output set-up time	5	7.2	-	ns	-
toH	Output hold time	5	9	-	ns	-
<b>Inputs CMD, DAT (referenced to CLK)</b>						
tIDLY	Input delay time during data transfer	-	-	13.7	ns	
tIH	Input hold time	2.5	-	-	ns	

**Figure 2-31** shows the CMD/DATA input/output timings of the eMMC in HS200 mode.

**Figure 2-31** CMD/DATA input/output timings of the eMMC in HS200 mode

[Table 2-47](#) describes the CMD/DATA input/output timing parameters of the eMMC in HS200 mode.

**Table 2-47** CMD/DATA input/output timing parameters in HS200 mode

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
<b>Clock CLK</b>						
$t_{PERIOD}$	Cycle time data transfer mode	6.73	6.73	-	ns	-
Duty Cycle	Clock duty cycle	30	50	70	%	-
<b>Outputs DAT (referenced to CLK)</b>						
$t_{OSU}$	Output set-up time	1.7	3.0	-	ns	
$t_{OH}$	Output hold time	1.1	1.9	-	ns	
<b>Inputs CMD, DAT (referenced to CLK)</b>						

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
t <sub>PH</sub>	Phase difference between component output CMD/DATA and RX CLK	0	-	2	UI	UI indicates the unit interval. For example, UI = 6.73 ns when the frequency is 148.5 MHz.
t <sub>VW</sub>	Input valid data window	0.575	-	-	UI	t <sub>VW</sub> = 3.87 ns when the frequency is 148.5 MHz

Figure 2-32 shows the data output timing of the eMMC in HS400 mode.

**Figure 2-32** Data output timing in HS400 mode

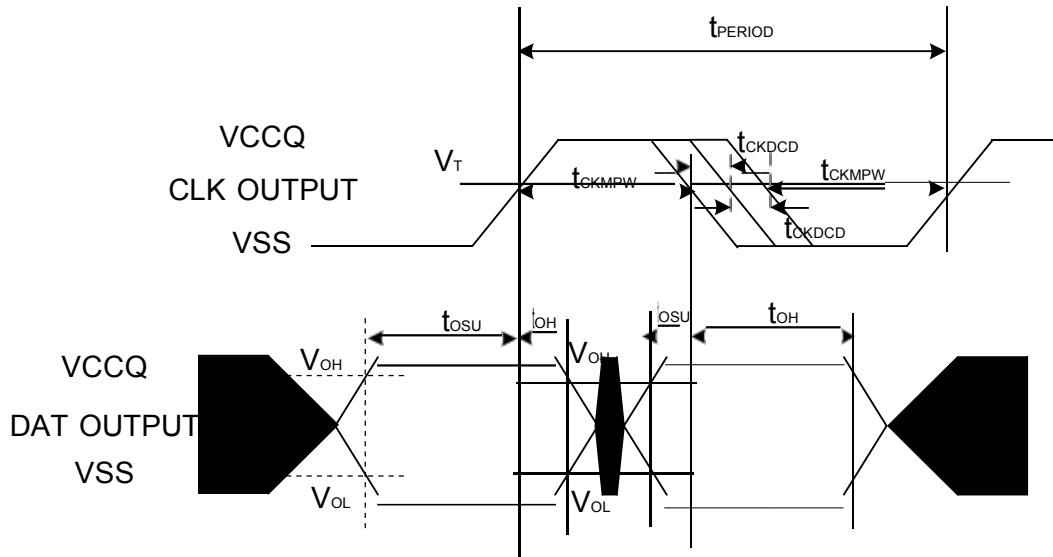


Table 2-48 describes the data output timing parameters of the eMMC in HS400 mode.

**Table 2-48** Data output timing parameters in HS400 mode

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
Clock CLK						
t <sub>PERIOD</sub>	Cycle time data transfer mode	6.73	6.73	-	ns	-

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
tCKMPW	Minimum pulse width	2.2	3.03	-	ns	-
Output DAT (referenced to CLK)						
tosUddr	Output set-up time	0.6	-	-	ns	
toHddr	Output hold time	0.6	-	-	ns	

Figure 2-33 shows the data input timing of the eMMC in HS400 mode.

**Figure 2-33** Data input timing in HS400 mode

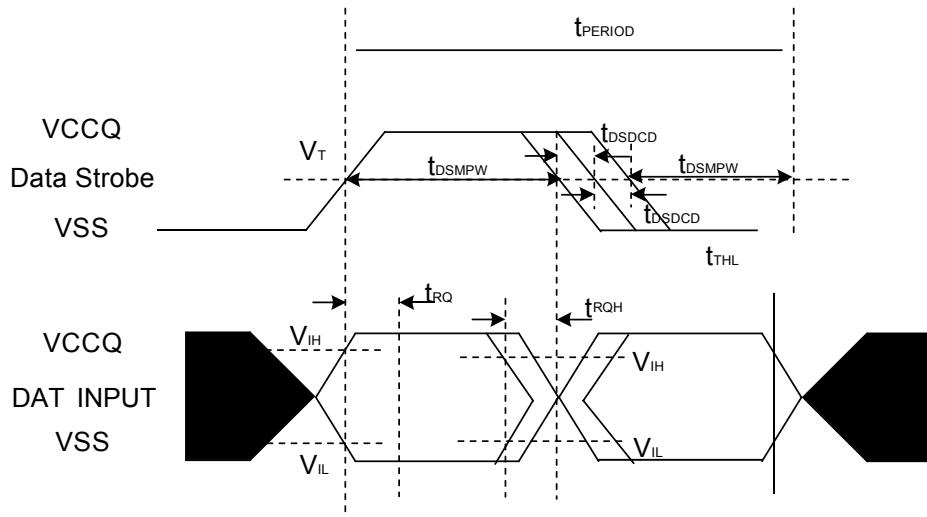


Table 2-49 describes the data input timing parameters of the eMMC in HS400 mode.

**Table 2-49** Data input timing parameters in HS400 mode

Symbol	Parameter	Min	Typ	Max	Unit
Data Strobe					
tPERIOD	Cycle time data transfer mode	6.73	-	-	-
tDSDCD	Duty cycle distortion	0.0	-	0.2	ns
tDSMPW	Minimum pulse width	2.0	-	-	ns
Input DAT (referenced to Data Strobe)					
tRQ	Input skew	-	-	0.4	ns

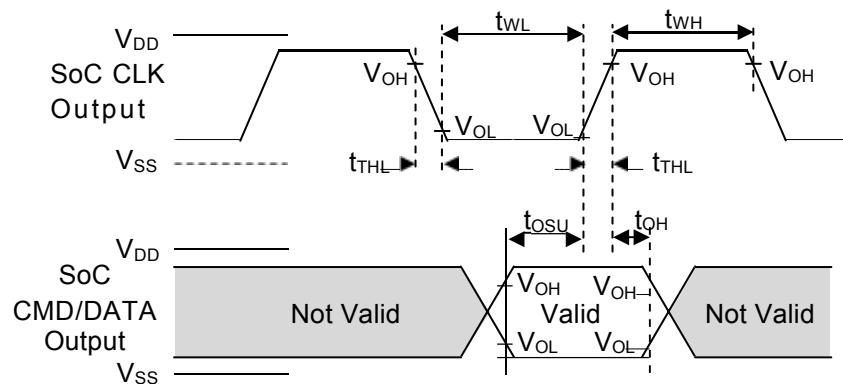


Symbol	Parameter	Min	Typ	Max	Unit
t <sub>RQH</sub>	Input holdskew	-	-	0.4	ns

## SDIO Interface Timings

[Figure 2-34](#) shows the CMD/DATA output timings of the SDIO in DS mode, and [Figure 2-35](#) shows the CMD/DATA input timings of the SDIO in DS mode. [Figure 2-36](#) shows the CMD/DATA output timings of the DS in HS mode. [Figure 2-37](#) shows the CMD/DATA input timings of the SDIO in HS mode.

**Figure 2-34** CMD/DATA output timings of the SDIO in DS mode



**Figure 2-35** CMD/DATA input timings of the SDIO in DS mode

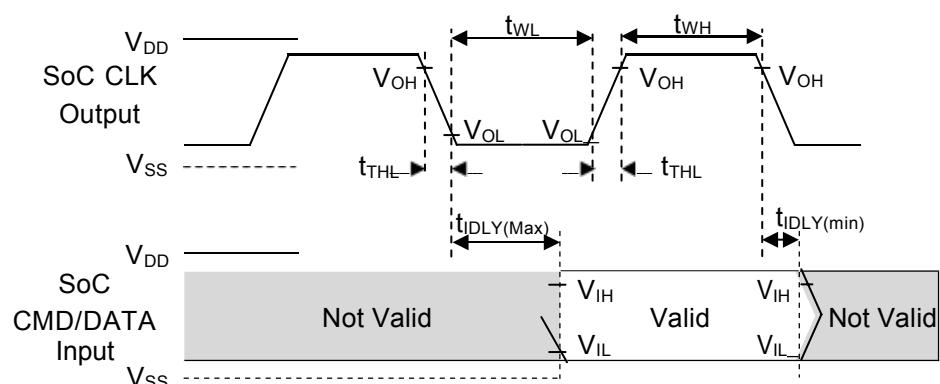




Figure 2-36 CMD/DATA output timings of the SDIO in HS mode

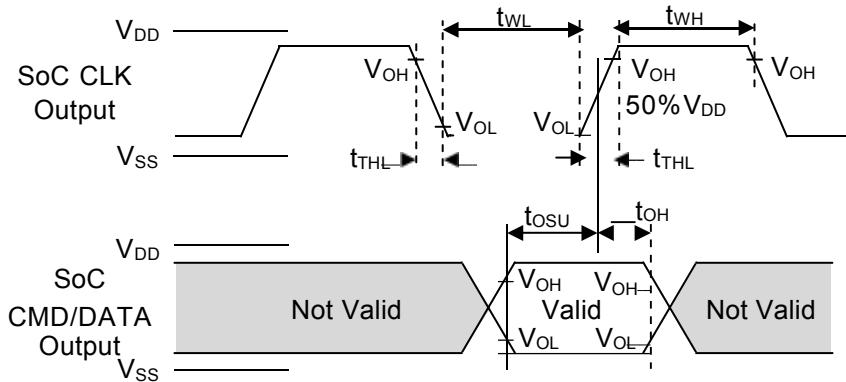


Figure 2-37 CMD/DATA input timings of the SDIO in HS mode

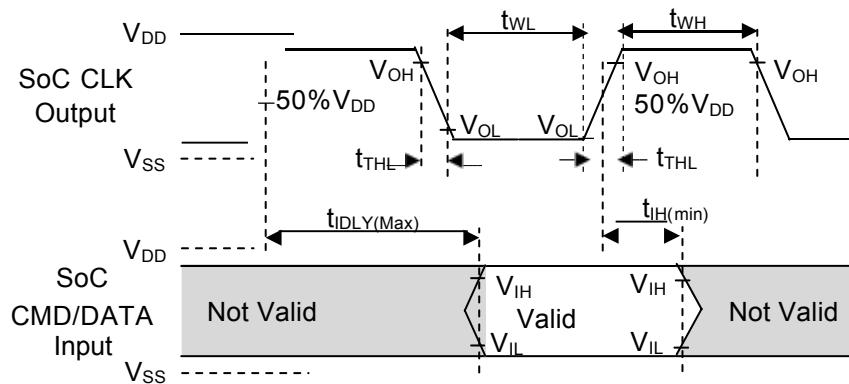


Table 2-50 shows the CMD/DATA input/output timings of the SDIO in DS mode.

Table 2-50 Input/output timings of the SDIO in DS mode

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
Clock CLK						
t <sub>CLK</sub>	Cycle time data transfer mode	40	40.4	-	ns	-
t <sub>WH</sub>	Clock high time	10	19	-	ns	-
t <sub>WL</sub>	Clock low time	10	19	-	ns	-
Outputs CMD/DAT (referenced to CLK)						
tosu	Output set-up time	5	18	-	ns	-



Symbol	Parameter	Min	Typ	Max	Unit	Remarks
t <sub>OH</sub>	Output hold time	5	18	-	ns	-
<b>Inputs CMD/DAT</b> (referenced to CLK)						
t <sub>IDLY</sub>	Input delay time during data transfer mode	0	-	14	ns	-
t <sub>IDLY</sub>	Input delay time during identification mode	0	-	50	ns	-

[Table 2-51](#) shows the CMD/DATA input/output timings of the SDIO in HS mode.

**Table 2-51** CMD/DATA input/output timing parameters of the SDIO in HS mode

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
Clock CLK						
t <sub>CLK</sub>	Cycle time data transfer mode	20	20.2	-	ns	-
t <sub>WH</sub>	Clock high time	7	7.89	-	ns	-
t <sub>WL</sub>	Clock low time	7	7.89	-	ns	-
<b>Outputs CMD/DAT</b> (referenced to CLK)						
t <sub>OSU</sub>	Output set-up time	6	8	-	ns	-
t <sub>OH</sub>	Output hold time	2	9	-	ns	-
<b>Inputs CMD/DAT</b> (referenced to CLK)						
t <sub>IDLY</sub>	Input delay time during data transfer mode	-	-	14	ns	-
t <sub>IH</sub>	Input hold time	2.5	-	-	ns	-



# Contents

<b>3 System.....</b>	<b>3-1</b>
3.1 Reset.....	3-1
3.1.1 Overview.....	3-1
3.1.2 Reset Control .....	3-1
3.1.3 Reset Configuration.....	3-3
3.2 Clock.....	3-4
3.2.1 Overview.....	3-4
3.2.2 Clock Control Block Diagram.....	3-4
3.2.3 Clock Distribution.....	3-5
3.2.4 CRG Register Summary.....	3-9
3.2.5 CRG Register Description.....	3-12
3.3 Processor Subsystem .....	3-51
3.4 Interrupt System .....	3-51
3.5 System Controller.....	3-53
3.5.1 Overview.....	3-53
3.5.2 Features .....	3-53
3.5.3 Function Description .....	3-54
3.5.4 System Controller Registers .....	3-54
3.5.5 SOC MISC1 Registers .....	3-66
3.5.6 SCO MISC2 Registers .....	3-73
3.6 DMA Controller.....	3-79
3.6.1 Overview.....	3-79
3.6.2 Features .....	3-79
3.6.3 Function Description .....	3-79
3.6.4 Working Mode .....	3-85
3.6.5 Register Summary.....	3-92
3.6.6 Register Description .....	3-95
3.7 Timer .....	3-111
3.7.1 Overview.....	3-111
3.7.2 Features .....	3-111



3.7.3 Function Description .....	3-112
3.7.4 Operating Mode .....	3-112
3.7.5 Register Summary .....	3-113
3.7.6 Register Description .....	3-114
3.8 Watchdog .....	3-118
3.8.1 Overview .....	3-118
3.8.2 Features .....	3-118
3.8.3 Function Description .....	3-118
3.8.4 Operating Mode .....	3-120
3.8.5 Register Summary .....	3-121
3.8.6 Register Description .....	3-121
3.9 RTC .....	3-124
3.9.1 Overview .....	3-124
3.9.2 Features .....	3-124
3.9.3 Function Description .....	3-125
3.9.4 Operating Mode .....	3-125
3.9.5 Register Summary .....	3-127
3.9.6 RTC Internal Register Description .....	3-129
3.10 Power Management and Low-Power Mode Control .....	3-145
3.10.1 Overview .....	3-145
3.10.2 Clock Gating and Clock Frequency Adjustment .....	3-145
3.10.3 Module Low-Power Control .....	3-146
3.10.4 DDR Low-Power Control .....	3-146
3.10.5 SVB Function Description .....	3-146
3.10.6 Tsensor_CTRL Function Description .....	3-146
3.10.7 Tsensor_CTRL Register Summary .....	3-147
3.10.8 Tsensor_CTRL Register Description .....	3-148



## Figures

<b>Figure 3-1</b> Diagram of controlling reset signals .....	3-2
<b>Figure 3-2</b> Internal reset and external reset.....	3-3
<b>Figure 3-3</b> Functional block diagram of the CRG .....	3-5
<b>Figure 3-4</b> Clock distribution diagram .....	3-6
<b>Figure 3-5</b> Formats of the linked list stored in the DDR.....	3-84
<b>Figure 3-6</b> Application block diagram of the watchdog .....	3-119



## Tables

<b>Table 3-1</b> Types of reset signals.....	3-2
<b>Table 3-2</b> Optional clocks of major modules .....	3-6
<b>Table 3-3</b> Summary of CRG registers (base address: 0x1101_0000) .....	3-9
<b>Table 3-4</b> Mapping between the interrupt bits and interrupt sources .....	3-51
<b>Table 3-5</b> Summary of system controller registers (base address: 0x1102_0000) .....	3-54
<b>Table 3-6</b> Summary of SOC MISC1 registers (base address: 0x1102_0000).....	3-66
<b>Table 3-7</b> Summary of SOC MISC2 registers (base address: 0x1795_0000).....	3-73
<b>Table 3-8</b> Description of peripheral single and burst hardware request line IDs for DMAC.....	3-80
<b>Table 3-9</b> Serial numbers of the last single hardware request lines received by DMAC peripherals .....	3-80
<b>Table 3-10</b> Description of access space for DMAC.....	3-81
<b>Table 3-11</b> Variable in the register offset addresses .....	3-92
<b>Table 3-12</b> DMAC registers (base address: 0x1028_0000).....	3-92
<b>Table 3-13</b> Summary of timer registers .....	3-114
<b>Table 3-14</b> Summary of watchdog registers (base address: 0x1103_0000) .....	3-121
<b>Table 3-15</b> Summary of internal RTC registers (base address: 0x1111_0000) .....	3-127
<b>Table 3-16</b> Summary of Tsensor_CTRL registers (base address: 0x1102_A000) .....	3-147



# 3 System

## 3.1 Reset

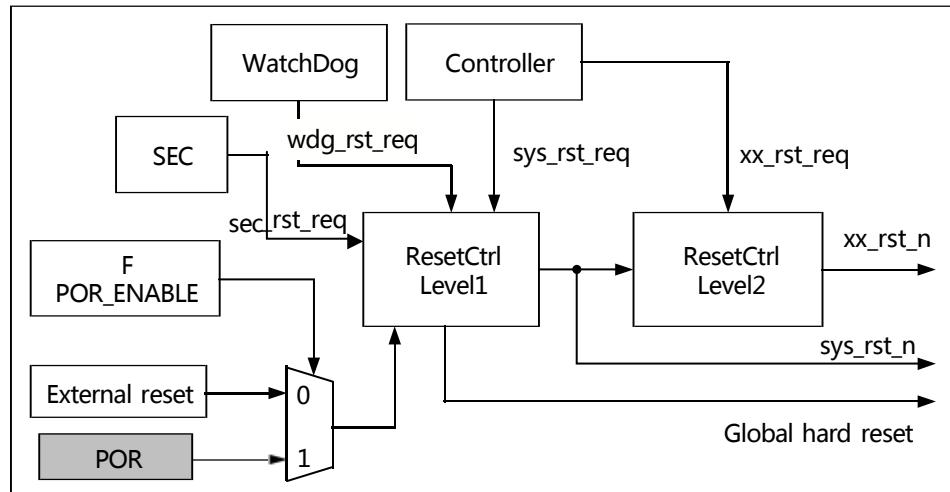
### 3.1.1 Overview

The reset management module manages the reset of the entire chip and each functional module. Specifically, the module:

- Manages and controls power-on reset.
- Manages and controls external reset.
- Manages and controls watchdog reset.
- Manages and controls secure system reset.
- Controls the system soft reset and the separate soft reset of each functional module.
- Synchronizes reset signals to the clock domain corresponding to each module.
- Generates reset signals of the functional modules in the chip.

### 3.1.2 Reset Control

[Figure 3-1](#) shows the diagram of controlling reset signals.

**Figure 3-1** Diagram of controlling reset signals

### NOTE

- POR: internal power-on reset (POR) module
- External reset: external reset signal from the pin
- POR\_ENABLE: whether the POR or external reset takes effect. This signal is derived from the pin.
- wdg\_rst\_req: watchdog reset request
- sec\_rst\_req: Secure system reset request
- sys\_rst\_req: global soft reset request signal. This signal is derived from the system controller.
- xx\_rst\_req: separate soft reset request signal of each submodule. This signal is derived from the clock and reset generator (CRG) system controller.
- xx\_rst\_n, sys\_rst\_n, and global hard reset: reset signals

Table 3-1 describes the types of reset signals.

**Table 3-1** Types of reset signals

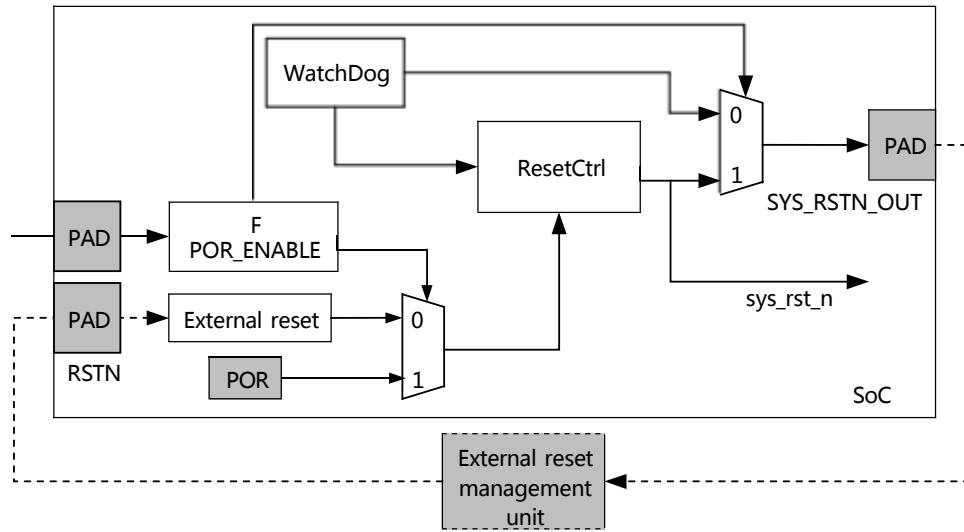
Type	Generation Mode	Function
Global hard reset signal	Watchdog reset request, secure system reset request, external reset, and internal POR module	Globally resets the SoC.
Global soft reset signal ( <code>syn_rst_n</code> )	Derived from the global soft reset register of the software configuration system controller	Resets all SoC modules except the clock reset circuit, test circuit, and some registers that cannot be soft-reset.



Type	Generation Mode	Function
Submodule reset signal (xx_rst_n)	Derived from the submodule reset control register of the CRG system controller	Separately resets each submodule of the chip.

Figure 3-2 shows the internal reset and external reset.

Figure 3-2 Internal reset and external reset



- When the POR\_ENABLE pin is connected to 1, the POR takes effect, but the external reset does not. The SYS\_RSTN\_OUT pin always outputs the reset signal sys\_rst\_n.
- When the POR\_ENABLE pin is connected to 0, the external reset takes effect, but the POR does not.

#### NOTICE

Hi3516CV610-10B/20B/20S/20G does not have external reset pins.

### 3.1.3 Reset Configuration

#### 3.1.3.1 Power-On Reset

To implement power-on reset, the following conditions must be met:

- The internal POR module generates a low-level pulse, and the low pulse duration is greater than 12 XIN crystal clock cycles.



- The clock input from the XIN pin of the crystal oscillator clock works properly.

### 3.1.3.2 System Reset

The system is reset in any of the following ways:

- Power-on reset
- External reset
- Watchdog reset
- Secure system reset
- Global soft reset, controlled by the system controller

### 3.1.3.3 Soft Reset

Soft reset control is implemented by configuring the corresponding CRG controller. For details about configurations, see the description of the reset register for each module.

#### NOTICE

- After a system soft reset request is sent, the entire chip must wait at least 20 ms to complete reset deassertion. No other system soft reset request can be sent within 20 ms. Otherwise, the system state will be out of order and the reset operation may fail to be completed.
- The separate soft reset of each module is not automatically deasserted. For example, if a module is reset after 1 is written to the related bit, the reset of this module is deasserted only when the related bit is set to 0.

## 3.2 Clock

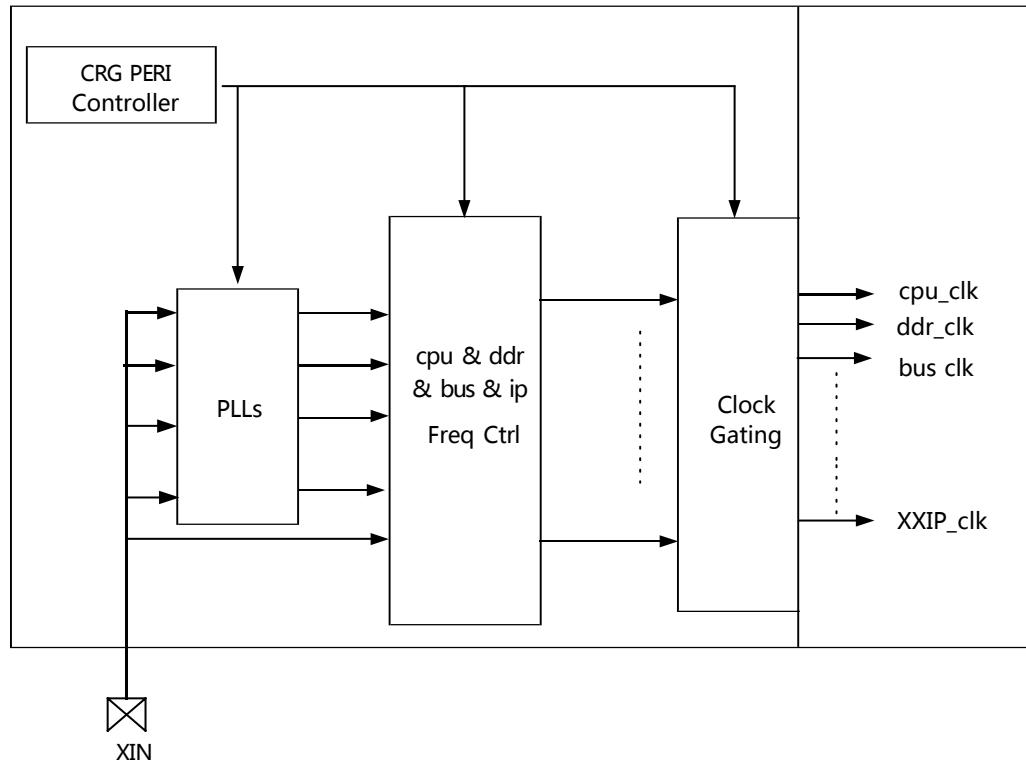
### 3.2.1 Overview

The clock management module manages the input, generation, and control of the chip clocks in a unified manner, and provides the following functions:

- Manages and controls clock inputs.
- Divides and controls clock frequencies.
- Generates working clocks for each module.

### 3.2.2 Clock Control Block Diagram

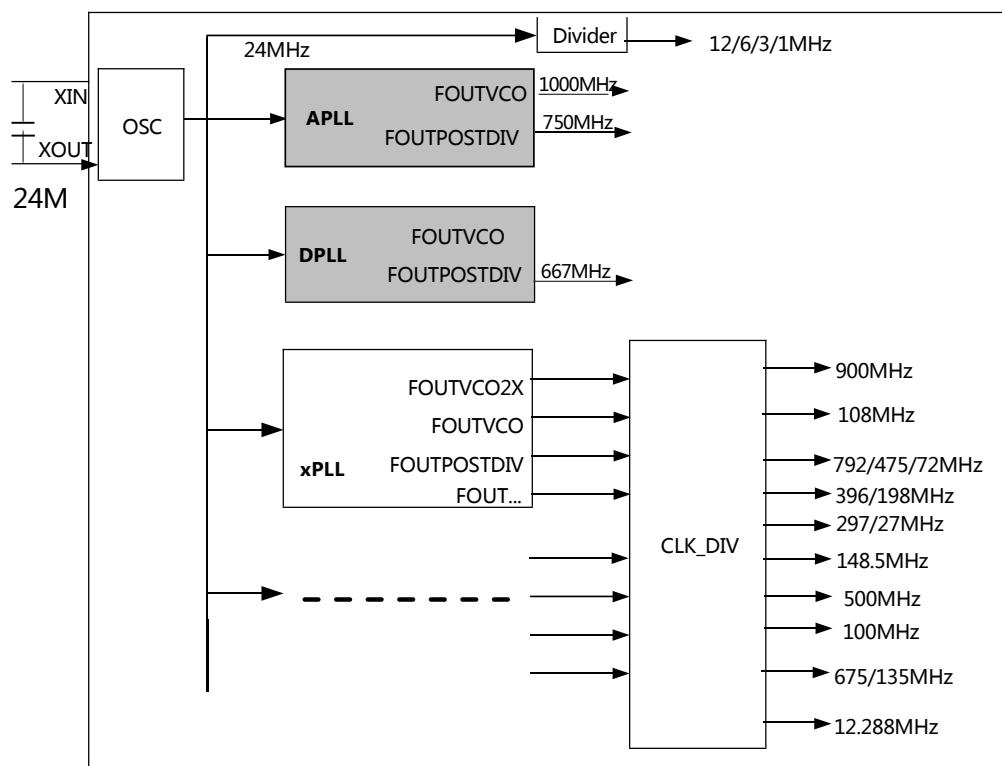
[Figure 3-3](#) shows the functional block diagram of the CRG.

**Figure 3-3** Functional block diagram of the CRG**NOTE**

XIN is the phase-locked loop (PLL) input clock and is always connected to a 24 MHz crystal.

### 3.2.3 Clock Distribution

The CRG configures, controls, and manages the internal PLLs and the input clocks from pins. It also generates the clocks required by each module. [Figure 3-4](#) shows the clock distribution diagram.

**Figure 3-4** Clock distribution diagram

**Table 3-2** lists major module clocks that can be selected. (For details about the configurations, see the description of the clock selection registers in each module.)

**Table 3-2** Optional clocks of major modules

Module	Default Frequency	Frequency	Remarks
A7	24 MHz	24 MHz	The maximum working clock cannot exceed 950 MHz.
		396 MHz	
		475 MHz	
		594 MHz	
		634 MHz	
		792 MHz	
		950 MHz	
CFG SYS	24 MHz	24 MHz	-
		100 MHz	-
DATA AXI	24 MHz	24 MHz	-
		198 MHz	



Module	Default Frequency	Frequency	Remarks
MIPI RX CTRL0/1	100 MHz	100 MHz	You can select any one of MIPI RX CTRL0/1.
		148.5 MHz	
		198 MHz	
		264 MHz	
		297 MHz	
		380 MHz	
		475 MHz	
VICAP PPC	-	100 MHz	Pixel Process Clock
		148.5 MHz	
		198 MHz	
		264 MHz	
		297 MHz	
		380 MHz	
		475 MHz	
VICAP ISP0	74.25 MHz	74.25 MHz	ISP Clock
		100 MHz	
		148.5 MHz	
		198 MHz	
		264 MHz	
		297 MHz	
		380 MHz (available only in offline mode)	
		475 MHz (available only in offline mode)	
AIAO	2376 MHz	2376 MHz	The actual operating frequency is the divided frequency of this clock.
		1900 MHz	
FMC	24 MHz	24 MHz	-
		100 MHz	-



Module	Default Frequency	Frequency	Remarks
		148.5 MHz	-
		198 MHz	Available in DDR mode and SDR mode 3.3 V only
		237.5 MHz	Available in DDR mode only
		264 MHz	Available in DDR mode only
		297 MHz	Available in DDR mode only
		396 MHz	Available in DDR mode 3.3 V only
SDIO0/eMMC	1600 KHz	1600 KHz	-
		100 MHz	
		198 MHz	
		396 MHz	
		594 MHz	
SDIO1	1600 KHz	1600 KHz	-
		100 MHz	
		198 MHz	
SENSOR0/1	74.25 MHz	74.25 MHz	Reference clock output from the SoC to the sensor
		72 MHz	
		54 MHz	
		50 MHz	
		37.125 MHz	
		36 MHz	
		27 MHz	
		25 MHz	
		24 MHz	
		12 MHz	



### 3.2.4 CRG Register Summary

Table 3-3 describes CRG registers.

**Table 3-3** Summary of CRG registers (base address: 0x1101\_0000)

Offset Address	Register	Description	Page Number
0x2000	PERI_CRG2048	SOC clock select register	<a href="#">3-12</a>
0x2020	PERI_CRG2056	SOC frequency configuration register	<a href="#">3-12</a>
0x2040	PERI_CRG2064	CPU frequency mode and reset configuration register	<a href="#">3-13</a>
0x2044	PERI_CRG2065	CPU_SUBSYS frequency indicator register	<a href="#">3-14</a>
0x204C	PERI_CRG2067	CPU CORE0/1 reset configuration register	<a href="#">3-15</a>
0x2050	PERI_CRG2068	CPU REE write permission control register	<a href="#">3-15</a>
0x2240	PERI_CRG2192	CPU debugging (CoreSight) clock configuration register	<a href="#">3-15</a>
0x2280	PERI_CRG2208	DDR clock and reset configuration register	<a href="#">3-16</a>
0x2A80	PERI_CRG2720	DMAC clock and soft reset control register	<a href="#">3-16</a>
0x35C0	PERI_CRG3440	SDIO0/eMMC clock and reset control register	<a href="#">3-17</a>
0x35C4	PERI_CRG3441	SDIO0/eMMC P4 DLL control register.	<a href="#">3-18</a>
0x35C8	PERI_CRG3442	SDIO0/eMMC DRV DLL control register.	<a href="#">3-19</a>
0x35CC	PERI_CRG3443	SDIO0/eMMC SAM DLL control register.	<a href="#">3-20</a>
0x35D0	PERI_CRG3444	SDIO0/eMMC DS DLL control register.	<a href="#">3-21</a>
0x35D4	PERI_CRG3445	SDIO0/eMMC edge detection clock phase select register	<a href="#">3-23</a>
0x35D8	PERI_CRG3446	SDIO0/eMMC status register	<a href="#">3-23</a>



Offset Address	Register	Description	Page Number
0x36C0	PERI_CRG3504	SDIO1 clock and reset control register	<a href="#">3-24</a>
0x36C4	PERI_CRG3505	SDIO1 P4 DLL control register	<a href="#">3-25</a>
0x36C8	PERI_CRG3506	SDIO1 DRV DLL control register	<a href="#">3-25</a>
0x36CC	PERI_CRG3507	SDIO1 SAM DLL control register	<a href="#">3-27</a>
0x36D4	PERI_CRG3509	SDIO1 edge detection clock phase select register	<a href="#">3-28</a>
0x36D8	PERI_CRG3510	SDIO1 status register	<a href="#">3-28</a>
0x37CC	PERI_CRG3571	ETH clock and soft reset control register	<a href="#">3-29</a>
0x38C0	PERI_CRG3632	USB2.0 PHY clock and soft reset control register	<a href="#">3-30</a>
0x38C8	PERI_CRG3634	USB2.0 controller clock and soft reset control register	<a href="#">3-31</a>
0x3F40	PERI_CRG4048	FMC clock and soft reset control register	<a href="#">3-31</a>
0x3F44	PERI_CRG4049	FMC frequency indicator register	<a href="#">3-32</a>
0x4180	PERI_CRG4192	UART0 clock and reset control register	<a href="#">3-33</a>
0x4188	PERI_CRG4194	UART1 clock and reset control register	<a href="#">3-34</a>
0x4190	PERI_CRG4196	UART2 clock and reset control register	<a href="#">3-34</a>
0x4280	PERI_CRG4256	PC0 clock and reset control register	<a href="#">3-35</a>
0x4288	PERI_CRG4258	PC1 clock and reset control register	<a href="#">3-35</a>
0x4290	PERI_CRG4260	PC2 clock and reset control register	<a href="#">3-36</a>
0x4480	PERI_CRG4384	SSP0 clock and reset control register	<a href="#">3-37</a>
0x4488	PERI_CRG4386	SSP1 clock and reset control register	<a href="#">3-37</a>
0x4598	PERI_CRG4454	PWM1 clock and soft reset control register	<a href="#">3-37</a>



Offset Address	Register	Description	Page Number
0x46C0	PERI_CRG4528	LSADC clock and soft reset control register	<a href="#">3-38</a>
0x4740	PERI_CRG4560	TSENSOR_CTRL clock and soft reset control register	<a href="#">3-38</a>
0x4750	PERI_CRG4564	TIMER01 clock and soft reset control register	<a href="#">3-39</a>
0x4754	PERI_CRG4565	TIMER23 clock and soft reset control register	<a href="#">3-39</a>
0x4768	PERI_CRG4570	GPIO clock and soft reset control register	<a href="#">3-40</a>
0x8440	PERI_CRG8464	SENSOR0 clock and reset configuration register	<a href="#">3-41</a>
0x8460	PERI_CRG8472	SENSOR1 clock and reset configuration register	<a href="#">3-42</a>
0x8540	PERI_CRG8528	MIPI_RX CTRL clock and reset configuration register	<a href="#">3-43</a>
0x8560	PERI_CRG8536	MIPI_RX PIX0 clock and reset configuration register	<a href="#">3-44</a>
0x8580	PERI_CRG8544	MIPI_RX PIX1 clock and reset configuration register	<a href="#">3-45</a>
0x9140	PERI_CRG9296	VI clock and reset control register	<a href="#">3-45</a>
0x9144	PERI_CRG9297	VI reset state register	<a href="#">3-46</a>
0x9150	PERI_CRG9300	VI ISP0 clock and reset control register	<a href="#">3-46</a>
0x9154	PERI_CRG9301	VI ISP1 clock and reset control register	<a href="#">3-47</a>
0x9160	PERI_CRG9304	VI COMS0 clock and reset control register	<a href="#">3-47</a>
0x9164	PERI_CRG9305	VI Port 0 clock and reset control register	<a href="#">3-48</a>
0x9184	PERI_CRG9313	VI port 1 clock and reset control register	<a href="#">3-49</a>
0xA880	PERI_CRG10784	AIAO clock and reset control register	<a href="#">3-49</a>



Offset Address	Register	Description	Page Number
0xAA80	PERI_CRG10912	AUDIO CODEC clock reset control register	3-50

### 3.2.5 CRG Register Description

#### PERI\_CRG2048

PERI\_CRG2048 is the SOC clock select register.

Offset Address: 0x2000 Total Reset Value: 0x0000\_1000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	Reserved	0x000002
[10:8]	RW	dataaxi_cksel	DATAAXI clock select 000: 24 MHz 001: 198 MHz Other values: reserved	0x0
[7]	-	reserved	Reserved	0x0
[6:4]	RW	cfgsys_cksel	CFGSYS clock select 000: 24 MHz 001: 100 MHz Other values: reserved	0x0
[3:0]	-	reserved	Reserved	0x0

#### PERI\_CRG2056

PERI\_CRG2056 is the SOC frequency configuration register.

Offset Address: 0x2020 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	Reserved	0x00000
[11]	-	reserved	Reserved	0x0
[10:8]	RO	dataaxi_sc_seled	DATAAXI clock switchover completion indicator	0x0



Bits	Access	Name	Description	Reset
			000: 24 MHz 001: 198 MHz Other values: reserved	
[7]	-	reserved	Reserved	0x0
[6:4]	RO	cfgsys_sc_seled	CFGSYS clock switchover completion indicator 000: 24 MHz 001: 100 MHz Other values: reserved	0x0
[3:0]	-	reserved	Reserved	0x0

## PERI\_CRG2064

PERI\_CRG2064 is the CPU frequency mode and reset configuration register.

Offset Address: 0x2040 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	Reserved	0x000
[20:18]	RW	a7_cksel	A7 clock select 000: 24 MHz 001: 396 MHz 010: 475 MHz 011: 594 MHz 100: 634 MHz 101: 792 MHz 110: 950 MHz 111: Reserved	0x0
[17]	-	reserved	Reserved	0x0
[16]	RW	cs_srst_req	CS soft reset request 0: reset deasserted 1: reset	0x0
[15]	-	reserved	Reserved	0x0
[14]	RW	socdbg_srst_req	SOCDBG soft reset request	0x0



Bits	Access	Name	Description	Reset
			0: reset deasserted 1: reset	
[13]	-	reserved	Reserved	0x0
[12]	RW	l2_srst_req	L2 soft reset request 0: reset deasserted 1: reset	0x0
[11]	-	reserved	Reserved	0x0
[10]	RW	arm_dbg1_srst_req	ARM DBG1 soft reset request 0: reset deasserted 1: reset	0x0
[9:5]	-	reserved	Reserved	0x00
[4]	RW	arm_dbg0_srst_req	ARM DBG0 soft reset request 0: reset deasserted 1: reset	0x0
[3:0]	-	reserved	Reserved	0x0

## PERI\_CRG2065

PERI\_CRG2065 is the CPU\_SUBSYS frequency indicator register

Offset Address: 0x2044 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RO	a7_sc_seled	A7 core clock switchover status indicator 000: 24 MHz 001: 396 MHz 010: 475 MHz 011: 594 MHz 100: 634 MHz 101: 792 MHz 110: 950 MHz 111: Reserved	0x0



## PERI\_CRG2067

PERI\_CRG2067 is a CPU CORE0/1 reset configuration register.

Offset Address: 0x204C Total Reset Value: 0x0000\_0100

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	Reserved	0x0000000
[8]	RW	arm_core1_srst_rq	ARM CORE1 soft reset request (configurable for secure or non-secure access) 0: reset deasserted 1: reset	0x1
[7:5]	-	reserved	Reserved	0x0
[4]	RW	arm_core0_srst_rq	ARM CORE0 soft reset request (configurable for secure or non-secure access) 0: reset deasserted 1: reset	0x0
[3:0]	-	reserved	Reserved	0x0

## PERI\_CRG2068

PERI\_CRG2068 is a CPU REE write permission control register

Offset Address: 0x2050 Total Reset Value: 0x0000\_000A

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW	cpu_cfg_nonsec_wr_disable	CPU REE write permission configuration (fixed on the secure side) 0xA: allowed Other values: not allowed	0xA

## PERI\_CRG2192

PERI\_CRG2192 is the CPU debugging (CoreSight) clock configuration register.

Offset Address: 0x2240 Total Reset Value: 0x0100\_0000



Bits	Access	Name	Description	Reset
[31:25]	-	reserved	Reserved	0x00
[24]	RW	soc_cs_ckgate_bypass	SoC global gating of the debugging channel 0: not bypassed 1: automatic gating bypassed	0x1
[23:0]	-	reserved	Reserved	0x000000

## PERI\_CRG2208

PERI\_CRG2208 is the DDR clock and reset configuration register.

Offset Address: 0x2280 Total Reset Value: 0x0000\_0710

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	Reserved	0x000000
[10]	RW	ddr_apb_cken	DDR APB gating 0: disabled 1: enabled	0x1
[9]	RW	ddr_dfi_cken	DDRC dfi_clk gating 0: disabled 1: enabled	0x1
[8]	-	reserved	Reserved	0x1
[7:5]	-	reserved	Reserved	0x0
[4]	RW	ddr_ref_div_srst_rq	Soft reset request of the static frequency divider of the DDR REF clock 0: reset deasserted 1: reset	0x1
[3:0]	-	reserved	Reserved	0x0

## PERI\_CRG2720

PERI\_CRG2720 is the DMAC clock and soft reset control register.

Offset Address: 0x2A80 Total Reset Value: 0x0000\_0001



Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5]	RW	edma_axi_cken	DMAC AXI core clock gating 0: disabled 1: enabled	0x0
[4]	RW	edma_apb_cken	DMAC APB core clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	edma_srst_req	DMAC soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG3440

PERI\_CRG3440 is the SDIO0/eMMC clock and reset control register.

Offset Address: 0x35C0 Total Reset Value: 0x0007\_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26:24]	RW	sdio0_cksel	SDIO0/eMMC 4X clock select. The interface clock is divided by 4 of this clock.  000: 1.6 MHz 001: 100 MHz 010: 198 MHz 011: 396 MHz 100: 594 MHz  Other values: reserved	0x0
[23:19]	-	reserved	Reserved	0x00
[18]	RW	sdio0_mmc_tx_srst_req	SDIO0/eMMC TX soft reset request 0: reset deasserted 1: reset	0x1
[17]	RW	sdio0_mmc_rx_srst_req	SDIO0/eMMC RX soft reset request	0x1



Bits	Access	Name	Description	Reset
			0: reset deasserted 1: reset	
[16]	RW	sdio0_mmc_srst_req	SDIO0/eMMC soft reset request 0: reset deasserted 1: reset	0x1
[15:2]	-	reserved	Reserved	0x0000
[1]	RW	sdio0_mmc_hclk_cken	SDIO0/eMMC bus clock gating 0: disabled 1: enabled	0x0
[0]	RW	sdio0_mmc_cken	SDIO0/eMMC clock gating 0: disabled 1: enabled	0x0

## PERI\_CRG3441

PERI\_CRG3441 is the SDIO0/eMMC P4 DLL control register.

Offset Address: 0x35C4 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	sdio0_mmc_dll_srst_req	Soft reset request of SDIO0/eMMC DLL (master and slave) 0: reset deasserted 1: reset	0x1
[0]	RW	sdio0_p4_dll_stop	Continuous dynamic tracking and detection enable for the DLL master 0: Dynamic detection is enabled and real-time phase calibration is performed based on detection parameters. 1: Dynamic detection is stopped after the DLL is locked. That is, the DLL is detected only once after initialization.	0x1



## PERI\_CRG3442

PERI\_CRG3442 is the SDIO0/eMMC DRV DLL control register.

Offset Address: 0x35C8 Total Reset Value: 0x0008\_0400

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19:15]	RW	sdio0_drv_clk_phase_sel	Clock phase select for SDIO0/eMMC DRV. The default value is 180°. 0x00: 0° 0x01: 11.25° 0x02: 22.5° ... 0x1E: 337.5° 0x1F: 348.75°	0x10
[14:11]	RW	sdio0_drv_dll_tune	Clock phase fine-tuning for SDIO0/eMMC DRV DLL 0x0: no tuning 0x1: delay increased by 1 tap 0x2: delay increased by 2 taps 0x3: delay increased by 3 taps ... 0x7: delay increased by 7 taps 0x8: no tuning 0x9: delay decreased by 1 tap 0xA: delay decreased by 2 taps 0xB: delay decreased by 3 taps ... 0xF: delay decreased by 7 taps	0x0
[10]	RW	sdio0_drv_dll_slave_en	SDIO0/eMMC DRV DLL slave enable 0: disabled 1: enabled	0x1
[9]	RW	sdio0_drv_dll_bypass	Bypass select for the SDIO0/eMMC DRV DLL slave 0: normal mode. The phase of the output clock is shifted relative to the input clock. The phase-shifting degree is configured by related registers of the SDIO0 controller.	0x0



Bits	Access	Name	Description	Reset
			1: bypassed	
[8]	RW	sdio0_drv_dll_mode	Mode select for the SDIO0/eMMC DRV DLL slave 0: normal mode; 1: The tap count of the DRV slave line is controlled by drv_dll_sel.	0x0
[7:0]	RW	sdio0_drv_dll_ssel	Tap count select for the SDIO0/eMMC DRV DLL slave line. Bit[7:6] are reserved. 0x00: Set 1-tap delay cell for the DLL slave. 0x01: Set 1-tap delay cell for the DLL slave. 0x02: Set 2-tap delay cells for the DLL slave. ... 0x3F: Set 63-tap delay cells for the DLL slave.	0x00

## PERI\_CRG3443

PERI\_CRG3443 is the SDIO0/eMMC SAM DLL control register.

Offset Address: 0x35CC Total Reset Value: 0x0000\_0400

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:11]	RW	sdio0_sam_dll_tune	Clock phase fine-tuning for SDIO0/eMMC SAM DLL 0x0: no tuning 0x1: delay increased by 1 tap 0x2: delay increased by 2 taps 0x3: delay increased by 3 taps ... 0x7: delay increased by 7 taps 0x8: no tuning 0x9: delay decreased by 1 tap 0xA: delay decreased by 2 taps	0x0



Bits	Access	Name	Description	Reset
			0xB: delay decreased by 3 taps ... 0xF: delay decreased by 7 taps	
[10]	RW	sdio0_sam_dll_slave_en	SDIO0/eMMC SAM DLL slave enable 0: disabled 1: enabled	0x1
[9]	-	reserved	Reserved	0x0
[8]	RW	sdio0_sam_dll_mode	Mode select for the SDIO0/eMMC SAM DLL slave 0: normal mode; 1: The tap count of the SAM slave line is controlled by sam_dll_ssel.	0x0
[7:0]	RW	sdio0_sam_dll_ssel	Tap count select for the SDIO0/eMMC SAM DLL slave line. Bit[7:6] are reserved. 0x00: Set 1-tap delay cell for the DLL slave. 0x01: Set 1-tap delay cell for the DLL slave. 0x02: Set 2-tap delay cells for the DLL slave. ... 0x3F: Set 63-tap delay cells for the DLL slave.	0x00

## PERI\_CRG3444

PERI\_CRG3444 is the SDIO0/eMMC DS DLL control register.

Offset Address: 0x35D0 Total Reset Value: 0x0000\_0400

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:11]	RW	sdio0_ds_dll_tune	Clock phase fine-tuning for SDIO0/eMMC DS DLL 0x0: no tuning 0x1: delay increased by 1 tap 0x2: delay increased by 2 tap	0x0



Bits	Access	Name	Description	Reset
			0x3: delay increased by 3 tap ... 0x7: delay increased by 7 tap 0x8: no tuning 0x9: delay decreased by 1 tap 0xA: delay decreased by 2 taps 0xB: delay decreased by 3 taps ... 0xF: delay decreased by 7 taps	
[10]	RW	sdio0_ds_dll_slave_en	SDIO0/eMMC DS DLL slave enable 0: disabled 1: enabled	0x1
[9]	RW	sdio0_ds_dll_bypass	SDIO0/eMMC DS DLL slave bypass select 0: normal mode. The phase of the output clock is shifted relative to the input clock. The phase-shifting degree is configured by related registers of the SDIO controller. 1: DRV DLL slave bypassed.	0x0
[8]	RW	sdio0_ds_dll_mode	Mode select for the SDIO0/eMMC DS DLL slave 0: normal mode; 1: The tap count of the DS slave line is controlled by ds_dll_ssel.	0x0
[7:0]	RW	sdio0_ds_dll_ssel	Tap count select for the SDIO0/eMMC DS DLL slave line. Bit[7:6] are reserved. 0x00: Set 1-tap delay cell for the DLL slave. 0x01: Set 1-tap delay cell for the DLL slave. 0x02: Set 2-tap delay cell for the DLL slave. ..... 0x3F: Set 63-tap delay cells for the DLL slave. .....	0x00



## PERI\_CRG3445

PERI\_CRG3445 is the SDIO0/eMMC edge detection clock phase select register.

Offset Address: 0x35D4 Total Reset Value: 0x0000\_0008

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4:0]	RW	sdio0_sample_b_cc_lk_sel	Clock phase (relative to the sample clock) select for edge detection 0x04: 45° 0x08: 90°(default value after power-on) ... 0x1C: 315° Note: Bit[1:0] must be 00.	0x08

## PERI\_CRG3446

PERI\_CRG3446 is the SDIO0/eMMC status register.

Offset Address: 0x35D8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	Reserved	0x0000
[16:13]	RO	reserved	Reserved	0x0
[12]	RO	sdio0_sam_dll_ready	SDIO0/eMMC sam DLL ready signal 0: The DLL slave is not ready. 1: The DLL slave is ready.	0x0
[11]	RO	sdio0_drv_dll_ready	SDIO0/eMMC drv DLL ready signal 0: The DLL slave is not ready. 1: The DLL slave is ready.	0x0
[10]	RO	sdio0_ds_dll_ready	SDIO0/eMMC DS DLL ready signal 0: The DLL slave is not ready. 1: The DLL slave is ready.	0x0
[9]	RO	sdio0_p4_dll_locked	SDIO0/eMMC P4 DLL lock signal 0: The DLL master is unlocked. 1: The DLL master is locked.	0x0
[8]	RO	sdio0_p4_dll_overflow	SDIO0/eMMC P4 DLL overflow signal	0x0



Bits	Access	Name	Description	Reset
		ow	0: The DLL master has no overflow. 1: The DLL master has overflow.	
[7:0]	RO	sdio0_p4_dll_mdly_tap	SDIO0/eMMC P4 DLL mdly_tap signal Number of taps for the DLL master line	0x00

## PERI\_CRG3504

PERI\_CRG3504 is the SDIO1 clock and reset control register.

Offset Address: 0x36C0 Total Reset Value: 0x0007\_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26:24]	RW	sdio1_cksel	SDIO1 4X clock select. The interface clock is divided by 4 of this clock.  000: 1.6 MHz 001: 100 MHz 010: 198 MHz  Other values: reserved	0x0
[23:19]	-	reserved	Reserved	0x00
[18]	RW	sdio1_mmc_tx_srst_req	SDIO1 TX soft reset request 0: reset deasserted 1: reset	0x1
[17]	RW	sdio1_mmc_rx_srst_req	SDIO1 RX soft reset request 0: reset deasserted 1: reset	0x1
[16]	RW	sdio1_mmc_srst_req	SDIO1 soft reset request 0: reset deasserted 1: reset	0x1
[15:2]	-	reserved	Reserved	0x0000
[1]	RW	sdio1_mmc_hclk_cken	SDIO1 bus clock gating 0: disabled 1: enabled	0x0
[0]	RW	sdio1_mmc_cken	SDIO1 clock gating	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	

## PERI\_CRG3505

PERI\_CRG3505 is the SDIO1 P4 DLL control register.

Offset Address: 0x36C4 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	sdio1_mmc_dll_srs_t_req	Soft reset request of SDIO1 DLL (master and slave) 0: reset deasserted 1: reset	0x1
[0]	RW	sdio1_p4_dll_stop	Continuous dynamic tracking and detection enable for the DLL master 0: Dynamic detection is enabled and real-time phase calibration is performed based on detection parameters. 1: Dynamic detection is stopped after the DLL is locked. That is, the DLL is detected only once after initialization.	0x1

## PERI\_CRG3506

PERI\_CRG3506 is the SDIO1 DRV DLL control register.

Offset Address: 0x36C8 Total Reset Value: 0x0008\_0400

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19:15]	RW	sdio1_drv_clk_phase_sel	Clock phase select for SDIO1 DRV. The default value is 180°. 0x00: 0° 0x01: 11.25° 0x02: 22.5° ...	0x10



Bits	Access	Name	Description	Reset
			0x1E: 337.5° 0x1F: 348.75°	
[14:11]	RW	sdio1_drv_dll_tune	Clock phase fine-tuning for SDIO1 DRV DLL  0x0: no tuning 0x1: delay increased by 1 tap 0x2: delay increased by 2 taps 0x3: delay increased by 3 taps ... 0x7: delay increased by 7 taps 0x8: no tuning 0x9: delay decreased by 1 tap 0xA: delay decreased by 2 taps 0xB: delay decreased by 3 taps ... 0xF: delay decreased by 7 taps	0x0
[10]	RW	sdio1_drv_dll_slave_en	SDIO1 DRV DLL slave enable 0: disabled 1: enabled	0x1
[9]	RW	sdio1_drv_dll_bypass	Bypass select for the SDIO1 DRV DLL slave 0: normal mode. The phase of the output clock is shifted relative to the input clock. The phase-shifting degree is configured by related registers of the SDIO1 controller. 1: bypassed	0x0
[8]	RW	sdio1_drv_dll_mode	Mode select for the SDIO1 DRV DLL slave 0: normal mode 1: The tap count of the DRV slave line is controlled by drv_dll_sel.	0x0
[7:0]	RW	sdio1_drv_dll_ssel	Tap count select for the SDIO1 DRV DLL slave line. Bit[7:6] are reserved. 0x00: Set 1-tap delay cell for the DLL slave. 0x01: Set 1-tap delay cell for the DLL slave.	0x00



Bits	Access	Name	Description	Reset
			0x02: Set 2-tap delay cells for the DLL slave. ... 0x3F: Set 63-tap delay cells for the DLL slave.	

## PERI\_CRG3507

PERI\_CRG3507 is the SDIO1 SAM DLL control register.

Offset Address: 0x36CC Total Reset Value: 0x0000\_0400

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:11]	RW	sdio1_sam_dll_tune	Clock phase fine-tuning for SDIO1 SAM DLL 0x0: no tuning 0x1: delay increased by 1 tap 0x2: delay increased by 2 taps 0x3: delay increased by 3 taps ... 0x7: delay increased by 7 taps 0x8: no tuning 0x9: delay decreased by 1 tap 0xA: delay decreased by 2 taps 0xB: delay decreased by 3 taps ... 0xF: delay decreased by 7 taps	0x0
[10]	RW	sdio1_sam_dll_slave_en	SDIO1 SAM DLL slave enable 0: disabled 1: enabled	0x1
[9]	-	reserved	Reserved	0x0
[8]	RW	sdio1_sam_dll_mode	Mode select for the SDIO1 SAM DLL slave 0: normal mode; 1: The tap count of the SAM slave line is controlled by sam_dll_ssel.	0x0



Bits	Access	Name	Description	Reset
[7:0]	RW	sdio1_sam_dll_ssel	Tap count select for the SDIO1 SAM DLL slave line. Bit[7:6] are reserved. 0x00: Set 1-tap delay cell for the DLL slave. 0x01: Set 1-tap delay cell for the DLL slave. 0x02: Set 2-tap delay cells for the DLL slave. ... 0x3F: Set 63-tap delay cells for the DLL slave.	0x00

## PERI\_CRG3509

PERI\_CRG3509 is the SDIO1 edge detection clock phase select register.

Offset Address: 0x36D4 Total Reset Value: 0x0000\_0008

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4:0]	RW	sdio1_sample_b_cc_lk_sel	Clock phase (relative to the sample clock) select for edge detection 0x04: 45° 0x08: 90°(default value after power-on) ... 0x1C: 315° Note: Bit[1:0] must be 00.	0x08

## PERI\_CRG3510

PERI\_CRG3510 is the SDIO1 status register.

Offset Address: 0x36D8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12]	RO	sdio1_sam_dll_ready	SDIO1 SAM DLL ready signal	0x0



Bits	Access	Name	Description	Reset
			0: The DLL slave is not ready. 1: The DLL slave is ready.	
[11]	RO	sdio1_drv_dll_ready	SDIO1 DRV DLL ready signal 0: The DLL slave is not ready. 1: The DLL slave is ready.	0x0
[10]	-	reserved	Reserved	0x0
[9]	RO	sdio1_p4_dll_locked	SDIO1 P4 DLL lock signal 0: The DLL master is unlocked. 1: The DLL master is locked.	0x0
[8]	RO	sdio1_p4_dll_overflow	SDIO1 P4 DLL overflow signal 0: The DLL master has no overflow. 1: The DLL master has overflow.	0x0
[7:0]	RO	sdio1_p4_dll_mdly_tap	SDIO1 P4 DLL mdly_tap signal Number of taps for the DLL master line	0x00

## PERI\_CRG3571

PERI\_CRG3571 is the ETH clock and soft reset control register.

Offset Address: 0x37CC Total Reset Value: 0x0000\_0009

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7]	RW	ethcore_cksel	ETH clock source select 0: 61 MHz 1: 100 MHz	0x0
[6:4]	-	reserved	Reserved	0x0
[3]	RW	fephy_srst_req	FEPHY soft reset request 0: reset deasserted 1: reset	0x1
[2]	RW	fephy_cken	FEPHY clock gating 0: disabled 1: enabled	0x0
[1]	RW	eth_cken	ETH clock gating	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[0]	RW	hrst_eth_s	ETH soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG3632

PERI\_CRG3632 is the USB2.0 PHY clock and soft reset control register.

Offset Address: 0x38C0 Total Reset Value: 0x0000\_0033

Bits	Access	Name	Description	Reset
[31:6]	RO	reserved	Reserved	0x00000000
[5]	RW	usb2_phy_ref_pll_cken	Reference clock select for the USB2.0 PHY0 PLL 0: disabled 1: enabled	0x1
[4]	RW	usb2_phy_ref_xtal_cken	Reference clock gating for the USB2.0 PHY0 XTAL 0: disabled 1: enabled	0x1
[3]	RO	reserved	Reserved	0x0
[2]	RW	usb2_phy_apb_RST_req	Soft reset request of USB2.0 PHY0 0: reset deasserted 1: reset	0x0
[1]	RW	usb2_phy_utmi_RST_req	Digital soft reset request of USB2.0 PHY0 0: reset deasserted 1: reset	0x1
[0]	RW	usb2_phy_RST_req	Global soft reset request of USB2.0 PHY0 0: reset deasserted 1: reset	0x1



## PERI\_CRG3634

PERI\_CRG3634 is the USB2.0 controller clock and soft reset control register.

Offset Address: 0x38C8 Total Reset Value: 0x0001\_0131

Bits	Access	Name	Description	Reset
[31:17]	RO	reserved	Reserved	0x0000
[16]	RW	usb2_freeclk_cksel	UTMI clock source select of the USB2.0 controller 0: UTMI clock 1: free 60 MHz clock	0x1
[15:9]	RO	reserved	Reserved	0x00
[8]	RW	usb2_utmi_cken	UTMI clock gating of the USB2.0 controller 0: disabled 1: enabled	0x1
[7:6]	RO	reserved	Reserved	0x0
[5]	RW	usb2_ref_cken	REF clock gating of the USB2.0 controller 0: disabled 1: enabled	0x1
[4]	RW	usb2_bus_cken	USB2.0 bus clock gating 0: disabled 1: enabled	0x1
[3:1]	RO	reserved	Reserved	0x0
[0]	RW	usb2_srst_req	Soft reset request of the USB2.0 controller 0: reset deasserted 1: reset	0x1

## PERI\_CRG4048

PERI\_CRG4048 is the FMC clock and soft reset control register.

Offset Address: 0x3F40 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000



Bits	Access	Name	Description	Reset
[14:12]	RW	fmc_cksel	FMC clock source select (In SDR mode, PHY_CLK_OUT is DIV2 of the original clock. In DDR mode, PHY_CLK_OUT is DIV4 of the original clock.)  000: 24 MHz 001: 100 MHz 010: 148.5 MHz 011: 198 MHz (available only in DDR mode, 3.3 V) 100: 237.5 MHz (available only in DDR mode) 101: 264 MHz (available only in DDR mode) 110: 297 MHz (available only in DDR mode) 111: 396 MHz (available only in DDR mode, 3.3 V)	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	fmc_cken	FMC clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	fmc_srst_req	FMC soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4049

PERI\_CRG4049 is the FMC frequency indicator register.

Offset Address: 0x3F44 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	Reserved	0x0000000
[10:8]	RO	fmc_sc_seled	FMC clock switchover completion indicator 000: 24 MHz	0x0



Bits	Access	Name	Description	Reset
			001: 100 MHz 010: 148.5 MHz 011: 198 MHz (available only in DDR mode, 3.3 V) 100: 237.5MHz (available only in DDR mode) 101: 264MHz (available only in DDR mode) 110: 297MHz (available only in DDR mode) 111: 396MHz (available only in DDR mode, 3.3 V)	
[7:0]	-	reserved	Reserved	0x00

## PERI\_CRG4192

PERI\_CRG4192 is the UART0 clock and reset control register.

Offset Address: 0x4180 Total Reset Value: 0x0000\_2001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13:12]	RW	uart0_cksel	UART0 clock select 00: 100 MHz 01: 50 MHz 10: 24 MHz 11: 3 MHz	0x2
[11:5]	-	reserved	Reserved	0x00
[4]	RW	uart0_cken	UART0 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	uart0_srst_req	UART0 soft reset request 0: reset deasserted 1: reset	0x1



## PERI\_CRG4194

PERI\_CRG4194 is the UART1 clock and reset control register.

Offset Address: 0x4188 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13:12]	RW	uart1_cksel	UART1 clock select 00: 100 MHz 01: 50 MHz 10: 24 MHz 11: 3 MHz	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	uart1_cken	UART1 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	uart1_srst_req	UART1 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4196

PERI\_CRG4196 is the UART2 clock and reset control register.

Offset Address: 0x4190 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13:12]	RW	uart2_cksel	UART2 clock select 00: 100 MHz 01: 50 MHz 10: 24 MHz 11: 3 MHz	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	uart2_cken	UART2 clock gating	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[3:1]	-	reserved	Reserved	0x0
[0]	RW	uart2_srst_req	UART2 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4256

PERI\_CRG4256 is the I<sup>2</sup>C0 clock and reset control register.

Offset Address: 0x4280 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12]	RW	i2c0_cksel	I <sup>2</sup> C0 clock select 0: 50 MHz 1: 100 MHz	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	i2c0_cken	I <sup>2</sup> C0 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	i2c0_srst_req	I <sup>2</sup> C0 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4258

PERI\_CRG4258 is the I<sup>2</sup>C1 clock and reset control register.

Offset Address: 0x4288 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000



Bits	Access	Name	Description	Reset
[12]	RW	i2c1_cksel	I <sup>2</sup> C1 clock select 0: 50 MHz 1: 100 MHz	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	i2c1_cken	I <sup>2</sup> C1 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	i2c1_srst_req	I <sup>2</sup> C1 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4260

PERI\_CRG4260 is the I<sup>2</sup>C2 clock and reset control register.

Offset Address: 0x4290 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12]	RW	i2c2_cksel	I <sup>2</sup> C2 clock select 0: 50 MHz 1: 100 MHz	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	i2c2_cken	I <sup>2</sup> C2 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	i2c2_srst_req	I <sup>2</sup> C2 soft reset request 0: reset deasserted 1: reset	0x1



## PERI\_CRG4384

PERI\_CRG4384 is the SSP0 clock and reset control register.

Offset Address: 0x4480 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	ssp0_cken	SSP0 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	ssp0_srst_req	SSP0 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4386

PERI\_CRG4386 is the SSP1 clock and reset control register.

Offset Address: 0x4488 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	ssp1_cken	SSP1 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	ssp1_srst_req	SSP1 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4454

PERI\_CRG4454 is the PWM1 clock and soft reset control register.

Offset Address: 0x4598 Total Reset Value: 0x0000\_0001



Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13:12]	RW	pwm1_cksel	PWM1 clock select 00: 1 MHz 01: 24 MHz 10: 198 MHz 11: reserved	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	pwm1_cken	PWM1 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	pwm1_srst_req	PWM1 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4528

PERI\_CRG4528 is the LSADC clock and soft reset control register.

Offset Address: 0x46C0 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	lsadc_cken	LSADC clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	lsadc_srst_req	LSADC soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG4560

PERI\_CRG4560 is the TSENSOR\_CTRL clock and soft reset control register.



Offset Address: 0x4740 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	tsensor_ctrl_cken	TSENSOR_CTRL working clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	tsensor_ctrl_pclk_ck_en	TSENSOR_CTRL APB clock gating 0: disabled 1: enabled	0x0

## PERI\_CRG4564

PERI\_CRG4564 is the TIMER01 clock and soft reset control register.

Offset Address: 0x4750 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	timer01_cken	TIMER01 clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	timer01_pclk_ck_en	TIMER01 APB clock gating 0: disabled 1: enabled	0x0

## PERI\_CRG4565

PERI\_CRG4565 is the TIMER23 clock and soft reset control register.

Offset Address: 0x4754 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000



[4]	RW	timer23_cken	TIMER23 working clock gating 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	timer23_pclk_cken	Clock gating of the TIMER23 APB 0: disabled 1: enabled	0x0

## PERI\_CRG4570

PERI\_CRG4570 is the GPIO clock and soft reset control register.

Offset Address: 0x4768 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:11]	-	reserved	Reserved	0x0000000
[10]	RW	gpio10_pclk_cken	GPIO10 APB clock gating 0: disabled 1: enabled	0x0
[9]	RW	gpio9_pclk_cken	GPIO9 APB clock gating 0: disabled 1: enabled	0x0
[8]	RW	gpio8_pclk_cken	GPIO8 APB clock gating 0: disabled 1: enabled	0x0
[7]	RW	gpio7_pclk_cken	GPIO7 APB clock gating 0: disabled 1: enabled	0x0
[6]	RW	gpio6_pclk_cken	GPIO6 APB clock gating 0: disabled 1: enabled	0x0
[5]	RW	gpio5_pclk_cken	GPIO5 APB clock gating 0: disabled 1: enabled	0x0
[4]	RW	gpio4_pclk_cken	GPIO4 APB clock gating	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[3]	RW	gpio3_pclk_cken	GPIO3 APB clock gating 0: disabled 1: enabled	0x0
[2]	RW	gpio2_pclk_cken	GPIO2 APB clock gating 0: disabled 1: enabled	0x0
[1]	RW	gpio1_pclk_cken	GPIO1 APB clock gating 0: disabled 1: enabled	0x0
[0]	RW	gpio0_pclk_cken	GPIO0 APB clock gating 0: disabled 1: enabled	0x1

## PERI\_CRG8464

PERI\_CRG8464 is a SENSOR0 clock and reset configuration register.

Offset Address: 0x8440 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	Reserved	0x0000
[16]	RW	clk_sensor0_pcrtl	Clock phase control of Sensor0 0: non inverted 1: inverted	0x0
[15:12]	RW	sensor0_cksel	Sensor0 clock select (reference clock output to the sensor by the chip) 0x0: 74.25 MHz 0x1: 72 MHz 0x2: 54 MHz 0x3: 50 MHz 0x4: 24 MHz 0x8: 37.125 MHz 0x9: 36 MHz	0x0



Bits	Access	Name	Description	Reset
			0xA: 27 MHz 0xB: 25 MHz 0xC: 12 MHz Other values: reserved	
[11:5]	-	reserved	Reserved	0x00
[4]	RW	sensor0_cken	Sensor0 clock gating (reference clock output to the sensor by the chip) 0: clock disabled 1: clock enabled	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RW	sensor0_ctrl_srst_req	Soft reset request of the sensor0 slave mode control module 0: reset deasserted 1: reset	0x1
[0]	RW	sensor0_srst_req	Sensor0 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG8472

PERI\_CRG8472 is a SENSOR1 clock and reset configuration register.

Offset Address: 0x8460 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	Reserved	0x0000
[16]	RW	clk_sensor1_pctrl	Clock phase control of Sensor1 0: non inverted 1: inverted	0x0
[15:12]	RW	sensor1_cksel	Sensor1 clock select (reference clock output to the sensor by the chip) 0x0: 74.25 MHz 0x1: 72 MHz 0x2: 54 MHz 0x3: 50 MHz	0x0



Bits	Access	Name	Description	Reset
			0x4: 24 MHz 0x8: 37.125 MHz 0x9: 36 MHz 0xA: 27 MHz 0xB: 25 MHz 0xC: 12 MHz Other values: reserved	
[11:5]	-	reserved	Reserved	0x00
[4]	RW	sensor1_cken	Sensor1 clock gating (reference clock output to the sensor by the chip) 0: clock disabled 1: clock enabled	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RW	sensor1_ctrl_srst_req	Soft reset request of the Sensor1 slave mode control module 0: reset deasserted 1: reset	0x1
[0]	RW	sensor1_srst_req	Sensor1 soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG8528

PERI\_CRG8528 is the MIPI\_RX CTRL clock and reset configuration register.

Offset Address: 0x8540 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	Reserved	0x000
[21]	RW	mipirx_hs1_pctrl	Phase control of the HS1 interface clock for the MIPI PHY 0 (default): non inverted 1: inverted	0x0
[20]	RW	mipirx_hs0_pctrl	Phase control of the HS0 interface clock for the MIPI PHY	0x0



Bits	Access	Name	Description	Reset
			0 (default): non inverted 1: inverted	
[19:6]	-	reserved	Reserved	0x0000
[5]	RW	mipirx_bus_cken	Bus logic clock gating of the MIPI 0: disabled 1: enabled	0x0
[4]	RW	cil_cken	CIL logic clock gating of the MIPI 0: disabled 1: enabled	0x0
[3]	RW	cbar_cken	MIPI CBAR clock gating 0: disabled 1: enabled	0x0
[2:1]	-	reserved	Reserved	0x0
[0]	RW	mipirx_bus_srst_rq	Bus logic soft reset request of the MIPI 0: reset deasserted 1: reset	0x1

## PERI\_CRG8536

PERI\_CRG8536 is the MIPI\_RX PIX0 clock and reset configuration register.

Offset Address: 0x8560 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	mipi_pix0_cken	Clock gating of MIPI PIX0 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	mipi_pix0_core_srs_t_req	Core logic soft reset request of MIPI PIX0 0: reset deasserted 1: reset	0x1



## PERI\_CRG8544

PERI\_CRG8544 is the MIPI\_RX PIX1 clock and reset configuration register.

Offset Address: 0x8580 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	mipi_pix1_cken	Clock gating of MIPI PIX1 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	mipi_pix1_core_srs_t_req	Core logic soft reset request of MIPI PIX1 0: reset deasserted 1: reset	0x1

## PERI\_CRG9296

PERI\_CRG9296 is the VICAP clock and reset control register.

Offset Address: 0x9140 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:12]	RW	vi_sc_cksel	VI SC clock select 000: reserved 001: 100 MHz 010: 148.5 MHz 011: 198 MHz 100: 264 MHz 101: 297 MHz 110: 380 MHz 111: 475 MHz	0x0
[11:6]	-	reserved	Reserved	0x00
[5]	RW	vi_bus_cken	BUS clock gating of the VI 0: disabled 1: enabled	0x0



Bits	Access	Name	Description	Reset
[4]	RW	vi_sc_cken	SC clock gating of the VI 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	vicap_srst_req	Soft reset request of the VICAP 0: reset deasserted 1: reset	0x1

## PERI\_CRG9297

PERI\_CRG9297 is the VICAP reset state register.

Offset Address: 0x9144 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	vicap_softrst_state	VICAP rest state 0: Reset is not complete. 1: Reset is complete and soft reset can be deasserted.	0x0

## PERI\_CRG9300

PERI\_CRG9300 is a VI ISP0 clock and reset control register.

Offset Address: 0x9150 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:12]	RW	isp0_cksel	ISP0 clock select 000: 74.25 MHz 001: 100 MHz 010: 148.5 MHz 011: 198 MHz 100: 264 MHz 101: 297 MHz	0x0



Bits	Access	Name	Description	Reset
			110: 380 MHz 111: 475 MHz	
[11:5]	-	reserved	Reserved	0x00
[4]	RW	isp0_cken	ISP0 clock gating 0: disabled 1: enabled	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RW	isp0_core_srst_req	ISP0 core soft reset request 0: reset deasserted 1: reset	0x1
[0]	RW	isp0_cfg_srst_req	ISP0 cfg soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG9301

PERI\_CRG9301 is a VI ISP1 clock and reset control register.

Offset Address: 0x9154 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:2]	-	reserved	Reserved	0x0
[1]	RW	isp1_core_srst_req	ISP1 core soft reset request 0: reset deasserted 1: reset	0x1
[0]	RW	isp1_cfg_srst_req	ISP1 cfg soft reset request 0: reset deasserted 1: reset	0x1

## PERI\_CRG9304

PERI\_CRG9304 is the VI CMOS0 clock and reset control register.

Offset Address: 0x9160 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:21]	-	reserved	Reserved	0x000
[20]	RW	vi_cmos_pctrl	Clock phase control of VI CMOS 0: non inverted 1: inverted	0x0
[19:0]	-	reserved	Reserved	0x00000

## PERI\_CRG9305

PERI\_CRG9305 is the VI port 0 clock and reset control register.

Offset Address: 0x9164 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:12]	RW	vi_p0_cksel	Clock select of MIPI CTRL Pixel0 000: 100 MHz 001: VI_PORT0(selected as the CMOS clock by setting vi_port0_input_sel of MISC_CTRL) 010: 148.5 MHz 011: 198 MHz 100: 264 MHz 101: 297 MHz 110: 380 MHz 111: 475 MHz	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	vi0_cken	Clock gating of VI port 0 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	vi_p0_srst_req	Soft reset request of VI port 0 0: reset deasserted 1: reset	0x1



## PERI\_CRG9313

PERI\_CRG9313 is the VI port 1 clock and reset control register.

Offset Address: 0x9184 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:12]	RW	vi_p1_cksel	Clock select of MIPI CTRL Pixel1 000: 100 MHz 001: VI_PORT1 (selected as the CMOS clock by setting vi_port1_input_sel of MISC_CTRL) 010: 148.5 MHz 011: 198 MHz 100: 264 MHz 101: 297 MHz 110: 380 MHz 111: 475 MHz	0x0
[11:5]	-	reserved	Reserved	0x00
[4]	RW	vi1_cken	Clock gating of VI port 1 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	vi_p1_srst_req	Soft reset request of VI port 1 0: reset deasserted 1: reset	0x1

## PERI\_CRG10784

PERI\_CRG10784 is the AIAO clock and reset control register.

Offset Address: 0xA880 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:13]	RW	aiao_cksel	Working clock select of the AIAO controller	0x0



Bits	Access	Name	Description	Reset
			00: 24 MHz 01: 50 MHz 10: 100 MHz 11: reserved	
[12]	RW	aiao_pll_cksel	Clock select of AIAO 0: 2376 MHz 1: 1900 MHz	0x0
[11:6]	-	reserved	Reserved	0x00
[5]	RW	aiao_cken	Clock gating of AIAO 0: disabled 1: enabled	0x0
[4]	RW	aiao_pll_cken	PLL clock gating of AIAO 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	aiao_srst_req	Soft reset request of AIAO 0: reset deasserted 1: reset	0x1

## PERI\_CRG10912

PERI\_CRG10912 is the AUDIO CODEC clock reset control register.

Offset Address: 0xAA80 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	codec_apb_cken	APB clock gating of AUDIO COEDC 0: disabled 1: enabled	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RW	codec_ana_rst_req	Analog reset request of the audio codec (The reset time must be at least 0.5 μs.) 0: reset deasserted	0x1



Bits	Access	Name	Description	Reset
			1: reset	
[0]	RW	codec_dig_RST_req	Digital logic reset request of the audio codec (The reset time must be at least 0.5 $\mu$ s.) 0: reset deasserted 1: reset	0x1

### 3.3 Processor Subsystem

The chip uses the dual-core Cortex-A7 processor. It has the following features:

- Supports the Working frequency of 950 MHz
- Supports the floating-point acceleration engine Neon.
- L1 cache contains 32 KB instruction cache (I-cache) and 32 KB data cache (D-cache).
- Supports 128 KB L2 cache shared by dual cores.
- Supports the memory management unit (MMU).
- Supports the Joint Test Action Group (JTAG) debugging interface.
- Integrates the generic interrupt controller (GIC), supporting 96 interrupt sources.

### 3.4 Interrupt System

The main CPU supports interrupt sources. [Table 3-4](#) describes the mapping between interrupt bits and interrupt sources.

**Table 3-4** Mapping between the interrupt bits and interrupt sources

Interrupt Vector Bit	Interrupt Source	Interrupt Vector Bit	Interrupt Source
0~31	Used internally in CPU	76	ETH
32	SOFTWARE	77	FEPHY
33	SDK_SOFTWARE	78	FEPHY_WOL
34	ALL_PLL_LOCK	79	Reserved
35	WDG	80	USB2



Interrupt Vector Bit	Interrupt Source	Interrupt Vector Bit	Interrupt Source
36	TIMER01	81	Reserved
37	TIMER23	82	SPACC_TEE
38	SEC_TIMER01	83	SPACC_REE
39	SEC_TIMER23	84	PKE_TEE
40	Reserved	85	PKE_REE
41	Reserved	86	RKP_TEE
42	UART0	87	RKP_REE
43	UART1	88	KLAD_TEE
44	UART2	89	KLAD_REE
45	Reserved	90	SEC_ALARM
46	Reserved	91~94	Reserved
47	I2C0	95	AIAO
48	I2C1	96	MIPI_RX
49	I2C2	97	VICAP_INT0
50	Reserved	98	VICAP_INT1
51	SSP0	99	VIPROC
52	SSP1	100	Reserved
53	Reserved	101	VEDU
54	Reserved	102	JPGE
55	GPIO0	103	VPSS
56	GPIO1	104	VGS
57	GPIO2	105	Reserved
58	GPIO3	106	Reserved
59	GPIO4	107	NPU
60	GPIO5	108	IVE
61	GPIO6	109	GZIP
62	GPIO7	110~112	Reserved
63	GPIO8	113	DDRT



Interrupt Vector Bit	Interrupt Source	Interrupt Vector Bit	Interrupt Source
64	GPIO9	114	DDRC_ERR
65	GPIO10	115	DDRC_SEC
66	Reserved	116	Reserved
67	TSENSOR	117	A7_AXIERR
68	LSADC	118	A7_COMMTX0
69	RTC	119	A7_COMMRX0
70	Reserved	120	A7_PMUIRQ0
71	EDMA	121	A7_COMMTX1
72	EDMA_NS	122	A7_COMMRX1
73	FMC	123	A7_PMUIRQ1
74	SDIO0	124~127	Reserved
75	SDIO1		

## 3.5 System Controller

### 3.5.1 Overview

The system controller manages the key functions of the system, and configures some functions of peripherals.

### 3.5.2 Features

The system controller has the following features:

- Monitors the system status.
- Provides general peripheral registers
- Provides write protection for key registers.
- Provides chip identification (ID) registers.



### 3.5.3 Function Description

#### 3.5.3.1 Soft Reset

The system controller supports global soft reset of the chip, which can be triggered by the **reboot** or **reset** command.

#### 3.5.3.2 Chip ID Register

The system controller provides a chip ID register CHIP\_ID, by which the software can identify the chip model.

### 3.5.4 System Controller Registers

#### 3.5.4.1 Register Summary

[Table 3-5](#)describes system controller registers.

**Table 3-5** Summary of system controller registers (base address: 0x1102\_0000)

Offset Address	Register	Description	Page Number
0x0014	SELF_BOOT_FLAG	System bootstrap control register	<a href="#">3-56</a>
0x0018	SYSSTAT	System status register	<a href="#">3-56</a>
0x001C	SOFTINT	Software interrupt register	<a href="#">3-57</a>
0x0020	SOFTTYPE	Software interrupt vector register	<a href="#">3-57</a>
0x0250	USE_POR_RREG0	The user dedicated register 0 that is resettable only by POR	<a href="#">3-58</a>
0x0254	USE_POR_RREG1	The user dedicated register 1 that is resettable only by POR	<a href="#">3-58</a>
0x0258	USE_POR_RREG2	The user dedicated register 2 that is resettable only by POR	<a href="#">3-58</a>
0x025C	USE_POR_RREG3	The user dedicated register 3 that is resettable only by POR	<a href="#">3-58</a>
0x0EE0	CHIP_ID	Chip ID register	<a href="#">3-59</a>
0x0EBC	VENDOR_ID	Vendor ID register	<a href="#">3-59</a>
0x1030	SDK_SOFTINT	Software interrupt register	<a href="#">3-59</a>
0x1034	SDK_SOFTTYPE	Software interrupt vector register	<a href="#">3-59</a>
0x1100	CUSTOMER_ID0	CUSTOMER_ID0 register	<a href="#">3-60</a>
0x1104	CUSTOMER_ID1	CUSTOMER_ID1 register	<a href="#">3-60</a>



Offset Address	Register	Description	Page Number
0x1108	CUSTOMER_ID2	CUSTOMER_ID2 register	<a href="#">3-60</a>
0x110C	CUSTOMER_ID3	CUSTOMER_ID3 register	<a href="#">3-61</a>
0x1200	CHIP_UNIQUE_ID0	CHIP_UNIQUE_ID0 register	<a href="#">3-61</a>
0x1204	CHIP_UNIQUE_ID1	CHIP_UNIQUE_ID1 register	<a href="#">3-61</a>
0x1208	CHIP_UNIQUE_ID2	CHIP_UNIQUE_ID2 register	<a href="#">3-61</a>
0x120C	CHIP_UNIQUE_ID3	CHIP_UNIQUE_ID3 register	<a href="#">3-62</a>
0x1210	CHIP_UNIQUE_ID4	CHIP_UNIQUE_ID4 register	<a href="#">3-62</a>
0x1214	CHIP_UNIQUE_ID5	CHIP_UNIQUE_ID5 register	<a href="#">3-62</a>
0x1300	USERREG0	User dedicated register 0	<a href="#">3-63</a>
0x1304	USERREG1	User dedicated register 1	<a href="#">3-63</a>
0x1308	USERREG2	User dedicated register 2	<a href="#">3-63</a>
0x130C	USERREG3	User dedicated register 3	<a href="#">3-63</a>
0x1310	USERREG4	User dedicated register 4	<a href="#">3-64</a>
0x1314	USERREG5	User dedicated register 5	<a href="#">3-64</a>
0x1318	USERREG6	User dedicated register 6	<a href="#">3-64</a>
0x131C	USERREG7	User dedicated register 7	<a href="#">3-64</a>
0x1320	USERREG8	User dedicated register 8	<a href="#">3-64</a>
0x1324	USERREG9	User dedicated register 9	<a href="#">3-65</a>
0x1328	USERREG10	User dedicated register 10	<a href="#">3-65</a>
0x132C	USERREG11	User dedicated register 11	<a href="#">3-65</a>
0x1330	USERREG12	User dedicated register 12	<a href="#">3-65</a>
0x1334	USERREG13	User dedicated register 13	<a href="#">3-66</a>
0x1338	USERREG14	User dedicated register 14	<a href="#">3-66</a>
0x133C	USERREG15	User dedicated register 15	<a href="#">3-66</a>



### 3.5.4.2 Register Description

#### SELF\_BOOT\_FLAG

SELF\_BOOT\_FLAG is the system bootstrap control register.

Offset Address: 0x0014 Total Reset Value: 0X0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	RO	softresreq	Reserved	0x00000
[7:0]	RO	self_boot_flag	System bootstrap mode select 0x00: Normal flash booting 0x01: UART burning to a non-bare chip 0x04: SD card burning to a non-bare chip 0x10: UART burning to a bare chip 0x11: UART burning to the DDR of a bare chip 0x20: USB device burning to a bare chip 0x40: SD card burning to a bare chip Other values: reserved	0x00

#### SYSSTAT

SYSSTAT is a system status register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0
[15]	RO	por_enable	Reset solution select 0: external reset 1: internal POR	0x0
[14:12]	-	reserved	Reserved	0x0
[11]	RO	sfc_emmc_boot_mode	Boot address mode select of the SPI NOR flash from which the system boots 0: 3-byte address mode 1: 4-byte address mode Boot mode select of the SPI NAND flash from which the system boots	0x0



Bits	Access	Name	Description	Reset
			0: 1-wire mode 1: 4-wire mode Boot mode select of the eMMC from which the system boots 0: 4-bit boot mode 1: 8-bit boot mode	
[10:5]	-	reserved	Reserved	0x00
[4]	RO	fast_boot_mode	Boot mode select 0: normal booting from the flash memory 1: burning over the serial port When the value is set to 0 or 1, select the boot mode or burning medium based on boot_sel[1:0].	0x0
[3:2]	RO	boot_sel	Boot medium select 00: booting from the SPI NOR flash 01: booting from the SPI NAND flash 11: booting from the eMMC Others: reserved	0x0
[1:0]	-	reserved	Reserved	0x0

## SOFTINT

SOFTINT is a software interrupt register.

Offset Address: 0x001C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	software_int	Writing 1 to this bit generates a software interrupt.	0x0

## SOFTTYPE

SOFTTYPE is a software interrupt vector register.

Offset Address: 0x0020 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	software_int_vector	Software interrupt vector	0x00000000

## USER\_POR\_REG0

USER\_POR\_REG0 is the user dedicated register 0 that is resettable only by POR. The register value is not cleared by the soft reset signal.

Offset Address: 0x0250 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_por_reg0	User dedicated register 0.	0x00000000

## USER\_POR\_REG1

USER\_POR\_REG1 is the user dedicated register 1 that is resettable only by POR. The register value is not cleared by the soft reset signal.

Offset Address: 0x0254 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_por_reg1	User dedicated register 1.	0x00000000

## USER\_POR\_REG2

USER\_POR\_REG2 is the user dedicated register 2 that is resettable only by POR. The register value is not cleared by the soft reset signal.

Offset Address: 0x0258 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_por_reg2	User dedicated register 2.	0x00000000

## USER\_POR\_REG3

USER\_POR\_REG3 is the user dedicated register 3 that is resettable only by POR. The register value is not cleared by the soft reset signal.



Offset Address: 0x025C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_por_reg3	User dedicated register 3.	0x00000000

## CHIP\_ID

CHIP\_ID is the chip ID register.

Offset Address: 0x0EE0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_id	Chip ID register (OTP readback value)	0x00000000

## VENDOR\_ID

VENDOR\_ID is the vendor ID register.

Offset Address: 0x0EEC Total Reset Value: 0x0000\_0035

Bits	Access	Name	Description	Reset
[31:0]	RO	vendor_id	Vendor ID register. The value is fixed at 0x35.	0x00000035

## SDK\_SOFTINT

SDK\_SOFTINT is a software interrupt register.

Offset Address: 0x1030 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	sdk_software_int	Writing 1 to this bit generates a software interrupt.	0x0

## SDK\_SOFTTYPE

SDK\_SOFTTYPE is a software interrupt vector register.

Offset Address: 0x1034 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	sdk_software_int_vector	Software interrupt vector.	0x00000000

## CUSTOMER\_ID0

CUSTOMER\_ID0 is the CUSTOMER\_ID0 register.

Offset Address: 0x1100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	customer_id0	Combination of CUSTOMER_ID0 to CUSTOMER_ID3, indicating by the combination of bit[127:0] CUSTOMER_ID[31:0]	0x00000000

## CUSTOMER\_ID1

CUSTOMER\_ID1 is the CUSTOMER\_ID1 register.

Offset Address: 0x1104 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	customer_id1	Combination of CUSTOMER_ID0 to CUSTOMER_ID3, indicating by the combination of bit[127:0] CUSTOMER_ID[63:32]	0x00000000

## CUSTOMER\_ID2

CUSTOMER\_ID2 is the CUSTOMER\_ID2 register.

Offset Address: 0x1108 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	customer_id2	Combination of CUSTOMER_ID0 to CUSTOMER_ID3, indicating by the combination of bit[127:0] CUSTOMER_ID[95:64]	0x00000000



## CUSTOMER\_ID3

CUSTOMER\_ID3 is the CUSTOMER\_ID3 register.

Offset Address: 0x110C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	customer_id3	Combination of CUSTOMER_ID0 to CUSTOMER_ID3, indicating by the combination of bit[127:0] CUSTOMER_ID[127:96]	0x00000000

## CHIP\_UNIQUE\_ID0

CHIP\_UNIQUE\_ID0 is a CHIP\_UNIQUE\_ID0 register.

Offset Address: 0x1200 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id0	Combination of CHIP_UNIQUE_ID0 to CHIP_UNIQUE_ID5, indicating by the combination of bit[191:0] CHIP_UNIQUE_ID[31:0]	0x00000000

## CHIP\_UNIQUE\_ID1

CHIP\_UNIQUE\_ID1 is a CHIP\_UNIQUE\_ID1 register.

Offset Address: 0x1204 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id1	Combination of CHIP_UNIQUE_ID0 to CHIP_UNIQUE_ID5, indicating by the combination of bit[191:0] CHIP_UNIQUE_ID[63:32]	0x00000000

## CHIP\_UNIQUE\_ID2

CHIP\_UNIQUE\_ID2 is a CHIP\_UNIQUE\_ID2 register.



Offset Address: 0x1208 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id2	Combination of CHIP_UNIQUE_ID0 to CHIP_UNIQUE_ID5, indicating by the combination of bit[191:0] CHIP_UNIQUE_ID[95:64]	0x00000000

## CHIP\_UNIQUE\_ID3

CHIP\_UNIQUE\_ID3 is a CHIP\_UNIQUE\_ID3 register.

Offset Address: 0x120C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id3	Combination of CHIP_UNIQUE_ID0 to CHIP_UNIQUE_ID5, indicating by the combination of bit[191:0] CHIP_UNIQUE_ID[127:96]	0x00000000

## CHIP\_UNIQUE\_ID4

CHIP\_UNIQUE\_ID4 is a CHIP\_UNIQUE\_ID4 register.

Offset Address: 0x1210 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id4	Combination of CHIP_UNIQUE_ID0 to CHIP_UNIQUE_ID5, indicating by the combination of bit[191:0] CHIP_UNIQUE_ID[159:128]	0x00000000

## CHIP\_UNIQUE\_ID5

CHIP\_UNIQUE\_ID5 is a CHIP\_UNIQUE\_ID5 register.

Offset Address: 0x1214 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	chip_unique_id5	Combination of CHIP_UNIQUE_ID0 to CHIP_UNIQUE_ID5, indicating by the	0x00000000



			combination of bit[191:0] CHIP_UNIQUE_ID[191:160]	
--	--	--	--	--

## USERREG0

USERREG0 is user dedicated register 0.

Offset Address: 0x1300 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg0	User dedicated register 0.	0x00000000

## USERREG1

USERREG1 is user dedicated register 1.

Offset Address: 0x1304 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg1	User dedicated register 1.	0x00000000

## USERREG2

USERREG2 is user dedicated register 2.

Offset Address: 0x1308 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg2	User dedicated register 2.	0x00000000

## USERREG3

USERREG3 is user dedicated register 3.

Offset Address: 0x130C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg3	User dedicated register 3.	0x00000000



## USERREG4

USERREG4 is user dedicated register 4.

Offset Address: 0x1310 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg4	User dedicated register 4.	0x00000000

## USERREG5

USERREG5 is user dedicated register 5.

Offset Address: 0x1314 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg5	User dedicated register 5.	0x00000000

## USERREG6

USERREG6 is user dedicated register 6.

Offset Address: 0x1318 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg6	User dedicated register 6.	0x00000000

## USERREG7

USERREG7 is user dedicated register 7.

Offset Address: 0x131C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg7	User dedicated register 7.	0x00000000

## USERREG8

USERREG8 is user dedicated register 8.

Offset Address: 0x1320 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg8	User dedicated register 8.	0x00000000

## USERREG9

USERREG9 is user dedicated register 9.

Offset Address: 0x1324 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg9	User dedicated register 9.	0x00000000

## USERREG10

USERREG10 is user dedicated register 10.

Offset Address: 0x1328 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg10	User dedicated register 10.	0x00000000

## USERREG11

USERREG11 is user dedicated register 11.

Offset Address: 0x132C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg11	User dedicated register 11.	0x00000000

## USERREG12

USERREG12 is user dedicated register 12.

Offset Address: 0x1330 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg12	User dedicated register 12.	0x00000000



## USERREG13

USERREG13 is user dedicated register 13.

Offset Address: 0x1334 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg13	User dedicated register 13.	0x00000000

## USERREG14

USERREG14 is user dedicated register 14.

Offset Address: 0x1338 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg14	User dedicated register 14.	0x00000000

## USERREG15

USERREG15 is user dedicated register 15.

Offset Address: 0x133C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	user_reg15	User dedicated register 15.	0x00000000

## 3.5.5 SOC MISC1 Registers

### 3.5.5.1 Register Summary

Table 3-6 describes the SOC MISC1 registers.

**Table 3-6** Summary of SOC MISC1 registers (base address: 0x1102\_0000)

Offset Address	Name	Description	Page
0x4130	DDRCA_REE_RANDOM_L	Lower 32-bit REE random value for the DDR scrambling function	<a href="#">3-67</a>



Offset Address	Name	Description	Page
0x4134	DDRCA_REE_RANDOM_H	Upper 32-bit REE random value for the DDR scrambling function	<a href="#">3-67</a>
0x4138	DDRCA_EN_REE	DDR scrambling mode control register	<a href="#">3-68</a>
0x413C	DDRCA_REE_UPDATE	DDR scrambling REE enable register	<a href="#">3-68</a>
0x4140	DDRCA_LOCK_REE	DDR scrambling random value lock register	<a href="#">3-68</a>
0x4144	DDRC_LOCK_CTRL1	DDRC lock register 1	<a href="#">3-69</a>
0x4148	DDRC_LOCK_CTRL2	DDRC lock register 2	<a href="#">3-69</a>
0x414C	DDRC_LOCK_CTRL3	DDRC lock register 3	<a href="#">3-69</a>
0x4150	DDRC_LOCK_CTRL4	DDRC lock register 4	<a href="#">3-70</a>
0x4154	DDRC_LOCK_CTRL5	DDRC lock register 5	<a href="#">3-70</a>
0x5000	SEC_CTRL1	Slave system security attribute control register 1	<a href="#">3-70</a>

### 3.5.5.2 Register Description

#### DDRCA\_REE\_RANDOM\_L

DDRCA\_REE\_RANDOM\_L is the lower 32-bit REE random value register for the DDR scrambling function.

Offset Address: 0x4130 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW_CLR	ddrca_ree_random_L	Bit[31:0] of the REE system random value for the DDR scrambling function. Writing 1 to ddrca_lock_ree clears this bit.	0x00000000

#### DDRCA\_REE\_RANDOM\_H

DDRCA\_REE\_RANDOM\_H is the upper 32-bit REE random value register for the DDR scrambling function.

Offset Address: 0x4134 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW_CLR	ddrca_ree_random_h	Bit[63:32] of the REE system random value for the DDR scrambling function. Writing 1 to ddrca_lock_ree clears this bit.	0x00000000

## DDRCA\_EN\_REE

DDRCA\_EN\_REE is the DDR scrambling mode control register.

Offset Address: 0x4138 Total Reset Value: 0x0000\_000A

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW_CLR	ddrca_en	DDR scrambling mode. Writing 1 to ddrca_lock_ree clears this bit. 0xA: The DDR scrambling function is disabled. 0x5: The DDR scrambling function is enabled. Other values: reserved	0xA

## DDRCA\_REE\_UPDATE

DDRCA\_REE\_UPDATE is the DDR scrambling REE enable register.

Offset Address: 0x413C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW_LOCK	ddrca_ree_update	Scrambling enable for the DDR REE random value. This bit takes effect after 1 is written. It is locked after 1 is written to ddrca_lock_ree.	0x0

## DDRCA\_LOCK\_REE

DDRCA\_LOCK\_REE is the DDR scrambling random value lock register.

Offset Address: 0x4140 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW_SEC_LOCK	ddrca_lock_ree	DDRC scrambling lock register. Writing 1 validates the lock. At the same time, ddrca_en, ddrca_ree_random_l, and ddrca_ree_random_h are cleared.	0x0

## DDRC\_LOCK\_CTRL1

DDRC\_LOCK\_CTRL1 is DDRC lock register 1.

Offset Address: 0x4144 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW_SEC_LOCK	ddrc_phy_write_lock	Lock control of the DDR PHY register 0: The register can be written. 1: The register cannot be written.	0x0

## DDRC\_LOCK\_CTRL2

DDRC\_LOCK\_CTRL2 is DDRC lock register 2.

Offset Address: 0x4148 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW_SEC_LOCK	ddrc_dmc_write_lock	Lock control of the DDR DMC register 0: The register can be written. 1: The register cannot be written.	0x0

## DDRC\_LOCK\_CTRL3

DDRC\_LOCK\_CTRL3 is DDRC lock register 3.

Offset Address: 0x414C Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW_SEC_LOCK	ddrc_qosbuf_write_lock	Lock control of the DDR QOSBUS register 0: The register can be written. 1: The register cannot be written.	0x0

## DDRC\_LOCK\_CTRL4

DDRC\_LOCK\_CTRL4 is DDRC lock register 4.

Offset Address: 0x4150 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW_SEC_LOCK	ddrc_axiif_write_lock	Lock control of the DDR AXIIF register 0: The register can be written. 1: The register cannot be written.	0x0

## DDRC\_LOCK\_CTRL5

DDRC\_LOCK\_CTRL5 is DDRC lock register 5.

Offset Address: 0x4154 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW_SEC_LOCK	ddrc_addr_atrb_write_lock	Lock control of the DDR ADDR ATRB register 0: The register can be written. 1: The register cannot be written.	0x0

## SEC\_CTRL1

SEC\_CTRL1 is slave system security attribute control register 1. (It is a secure register, which can be accessed only by the security module.)

Offset Address: 0x5000 Total Reset Value: 0x0001\_1111



Bits	Access	Name	Description	Reset
[31:19]	-	reserved	Reserved	0x0000
[18]	RW_SEC	i2c2_slv_sec_acc_ctrl	I <sup>2</sup> C2 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0
[17]	RW_SEC	i2c2_slv_sec_acc_ctrl_high	I <sup>2</sup> C2 read access control when i2c2_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0
[16]	RW_SEC	i2c2_slv_sec_acc_disable	I <sup>2</sup> C2 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1
[15]	-	reserved	Reserved	0x0
[14]	RW_SEC	uart1_slv_sec_acc_ctrl	UART1 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0
[13]	RW_SEC	uart1_slv_sec_acc_ctrl_high	UART1 read access control when uart1_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0
[12]	RW_SEC	uart1_slv_sec_acc_disable	UART1 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1
[11]	-	reserved	Reserved	0x0
[10]	RW_SEC	uart0_slv_sec_acc_ctrl	UART0 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0



Bits	Access	Name	Description	Reset
[9]	RW_SEC	uart0_slv_sec_acc_ctrl_high	UART0 read access control when uart0_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0
[8]	RW_SEC	uart0_slv_sec_acc_disable	UART0 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1
[7]	-	reserved	Reserved	0x0
[6]	RW_SEC	sec_timer23_slv_sec_acc_ctrl	sec_timer23 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0
[5]	RW_SEC	sec_timer23_slv_sec_acc_ctrl_high	sec_timer23 read access control when sec_timer23_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0
[4]	RW_SEC	sec_timer23_slv_sec_acc_disable	sec_timer23 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1
[3]	-	reserved	Reserved	0x0
[2]	RW_SEC	sec_timer01_slv_sec_acc_ctrl	sec_timer01 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0
[1]	RW_SEC	sec_timer01_slv_sec_acc_ctrl_high	sec_timer01 read access control when sec_timer01_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0



Bits	Access	Name	Description	Reset
[0]	RW_SEC	sec_timer01_slv_se c_acc_disable	sec_timer01 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1

## 3.5.6 SCO MISC2 Registers

### 3.5.6.1 Register Summary

Table 3-7 describes the SOC MISC2 registers.

**Table 3-7** Summary of SOC MISC2 registers (base address: 0x1795\_0000)

Offset Address	Name	Description	Page
0x000C	SPI_CFG10	SPI0/1 CS polarity configuration register	3-73
0x0018	I2C_MONTAGE_CFG0	I <sup>2</sup> C montage configuration register	3-74
0x0020	I2C_MONTAGE_CFG2	I <sup>2</sup> C montage read/write select register	3-74
0x0030	MIPI_RX_CFG	ISP working mode register	3-75
0x0034	MIPI_WORK_MODE	MIPI RX working mode control register	3-75
0x0038	VI_WORK_MODE	VI mode select register	3-76
0x800	SEC_CTRL2	Slave system security attribute control register 2	3-76

### 3.5.6.2 Register Description

#### SPI\_CFG10

SPI\_CFG10 is the SPI0/1 CS polarity configuration register

Offset Address: 0x000C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000



Bits	Access	Name	Description	Reset
[2]	RW	spi0_cs_mux	SPI0 CS signal 0: SPI0_CSNO output valid CS 1: SPI0_CSNI output valid CS	0x0
[1]	RW	spi0_cs_pctrl	Polarity control for the SPI0 CS signal 0: SPI0_CS, active low 1: SPI0_CS, active high	0x0
[0]	RW	spi1_cs_pctrl	Polarity control for the SPI1 CS signal 0: SPI1_CS, active low 1: SPI1_CS, active high	0x0

## I2C\_MONTAGE\_CFG0

I2C\_MONTAGE\_CFG0 is the I<sup>2</sup>C montage configuration register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1:0]	RW	i2c_montage_en	Montage enable for each I <sup>2</sup> C channel Bit[0]: I <sup>2</sup> C0 montage control Bit[1]: I <sup>2</sup> C1 montage control 0: disabled 1: enabled The I <sup>2</sup> C0 controller is used to control the I <sup>2</sup> C0 and I <sup>2</sup> C1 interfaces.	0x0

## I2C\_MONTAGE\_CFG2

I2C\_MONTAGE\_CFG2 is the I<sup>2</sup>C montage read/write select register.

Offset Address: 0x0020 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:4]	RW	i2c_write_slave	Writeback I <sup>2</sup> C select Bit[0]: I <sup>2</sup> C0	0x0



			Bit[1]: I <sup>2</sup> C1 0: disabled 1: enabled	
[3:2]	-	reserved	Reserved	0x0
[1:0]	RW	i2c_read_slave	Readback I <sup>2</sup> C select Bit[0]: I <sup>2</sup> C0 Bit[1]: I <sup>2</sup> C1 0: disabled 1: enabled	0x0

## MIPI\_RX\_CFG

MIPI\_RX\_CFG is the ISP operating mode configuration register.

Offset Address: 0x0030 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	Reserved	0x0
[29]	RW	viproc_vpss_online_mode	Operating mode select for VIPROC & VPSS 0: offline mode 1: online mode	0x0
[28]	RW	vicap_viproc_online_mode	Operating mode select for VICAP & VIPROC 0: offline mode 1: online mode	0x0
[27:0]	-	reserved	Reserved	0x00000000

## MIPI\_WORK\_MODE

MIPI\_WORK\_MODE is the MIPIRX operating mode configuration register.

Offset Address: 0x0034 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:4]	RW	mipi1_work_mode	Operating mode select for MIPI chn1	0x0



			00: MIPI mode 01: LVDS mode Other values: reserved	
[3:2]	-	reserved	Reserved	0x0
[1:0]	RW	mipi0_work_mode	Operating mode select for MIPI chn0 00: MIPI mode 01: LVDS mode Other values: reserved	0x0

## VI\_WORK\_MODE

VI\_WORK\_MODE is the VI mode select register.

Offset Address: 0x0038 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	Reserved	0x0000000
[8]	RW	vi_dvp_bt1120_input_sel	Function select of DVP/BT.1120 0: DVP mode 1: BT.1120 mode	0x0
[7:5]	-	reserved	Reserved	0x0
[4]	RW	vi_port1_input_sel	Function select of VI port 1 0: mipi 1: dvp/bt1120	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	vi_port0_input_sel	Function select of VI port 0 0: mipi 1: dvp/bt1120	0x0

## SEC\_CTRL2

SEC\_CTRL2 is slave system security attribute control register 2. (It is a secure register, which can be accessed only by the security module.)

Offset Address: 0x800 Total Reset Value: 0x0011\_1111



Bits	Access	Name	Description	Reset
[31:19]	-	reserved	Reserved	0x0002
[18]	RW_SEC	ssp1_slv_sec_acc_ctrl	ssp1 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0
[17]	RW_SEC	ssp1_slv_sec_acc_ctrl_high	ssp1 read access control when ssp1_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0
[16]	RW_SEC	ssp1_slv_sec_acc_disable	ssp1 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1
[15]	-	reserved	Reserved	0x0
[14]	RW_SEC	ssp0_slv_sec_acc_ctrl	ssp0 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0
[13]	RW_SEC	ssp0_slv_sec_acc_ctrl_high	ssp0 read access control when ssp0_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0
[12]	RW_SEC	ssp0_slv_sec_acc_disable	ssp0 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1
[11]	-	reserved	Reserved	0x0
[10]	RW_SEC	i2c1_slv_sec_acc_ctrl	I <sup>2</sup> C1 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0



Bits	Access	Name	Description	Reset
[9]	RW_SEC	i2c1_slv_sec_acc_ctrl_high	I <sub>C</sub> 1 read access control when i2c1_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0
[8]	RW_SEC	i2c1_slv_sec_acc_disable	I <sub>C</sub> 1 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1
[7]	-	reserved	Reserved	0x0
[6]	RW_SEC	i2c0_slv_sec_acc_ctrl	I <sub>C</sub> 0 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0
[5]	RW_SEC	i2c0_slv_sec_acc_ctrl_high	I <sub>C</sub> 0 read access control when i2c0_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0
[4]	RW_SEC	i2c0_slv_sec_acc_disable	I <sub>C</sub> 0 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1
[3]	-	reserved	Reserved	0x0
[2]	RW_SEC	uart2_slv_sec_acc_ctrl	UART2 secure access mode control 0: In secure mode, the read and write access is authenticated by the module. 1: In secure mode, the write operation supports only secure access.	0x0
[1]	RW_SEC	uart2_slv_sec_acc_ctrl_high	UART2 read access control when uart2_slv_sec_acc_ctrl is set to 1 0: In secure mode, the read access is authenticated by the module. 1: In secure mode, the read operation supports only secure access.	0x0



Bits	Access	Name	Description	Reset
[0]	RW_SEC	uart2_slv_sec_acc_disable	UART2 secure access mode disable 0: The secure mode is enabled. 1: The secure mode is disabled.	0x1

## 3.6 DMA Controller

### 3.6.1 Overview

The direction memory access (DMA) operation is a high-speed data transfer operation. It supports data read/write between peripherals and memories without using the CPU. The direction memory access controller (DMAC) directly transfers data between a memory and a peripheral, between peripherals, or between memories. This avoids the CPU intervention and reduces the interrupt handling overhead of the CPU.

### 3.6.2 Features

The DMAC has the following features:

- Provides four DMA channels. Each channel can be configured for a specific unidirectional transfer.
- Supports memory-to-memory, memory-to-peripherals, and peripherals-to-memory transfer.
- Supports DMAC flow control and peripheral flow control during peripheral transfer.
- Supports 32 sets of peripheral requests (each set includes the single, burst, and last single requests).
- Supports DMA transfer with the linked list.
- Allows the channel priority to be configured.
- Supports DMA burst length configured through software.
- Transfers data in 8-bit, 16-bit, 32-bit, or 64-bit mode.
- Supports only the little-endian mode.

### 3.6.3 Function Description

#### 3.6.3.1 Peripheral Request Lines

Table 3-8 describes the mapping between the single and burst hardware requests and the devices.

**Table 3-8** Description of peripheral single and burst hardware request line IDs for DMAC

Peripheral Hardware Request Line ID	Corresponding Device
0	DMA request of the I <sup>2</sup> C0 RX channel
1	DMA request of the I <sup>2</sup> C0 TX channel
2	DMA request of the I <sup>2</sup> C1 RX channel
3	DMA request of the I <sup>2</sup> C1 TX channel
4	DMA request of the I <sup>2</sup> C2 RX channel
5	DMA request of the I <sup>2</sup> C2 TX channel
6	DMA request of the SPI0 RX channel
7	DMA request of the SPI0 TX channel
8	DMA request of the SPI1 RX channel
9	DMA request of the SPI1 TX channel
10	DMA request of the UART0 RX channel
11	DMA request of the UART0 TX channel
12	DMA request of the UART1 RX channel
13	DMA request of the UART1 TX channel
14	DMA request of the UART2 RX channel
15	DMA request of the UART2 TX channel
16~31	Reserved

**Table 3-9** describes the mapping between the last single hardware requests and devices for the DMAC. The mapping takes effect only in peripheral flow control mode. When a peripheral sends a last single RX request, the peripheral requests the DMAC to trigger the last data transfer.

**Table 3-9** Serial numbers of the last single hardware request lines received by DMAC peripherals

Peripheral Hardware Request Line ID	Corresponding Device
0~9	Reserved
10	DMA request of the UART0 RX channel



Peripheral Hardware Request Line ID	Corresponding Device
11	Reserved
12	DMA request of the UART1 RX channel
13	Reserved
14	DMA request of the UART2 RX channel
15~31	Reserved

### 3.6.3.2 Access Space

Table 3-10 shows the address space that can be accessed by the DMAC.

**Table 3-10** Description of access space for DMAC

Access Space Type	Description
Memory	DDR space for non-secure attributes
	DDR space for secure attributes
	DDR space for shared attributes
Peripherals	P0
	P1
	P2
	SPI0
	SPI1
	UART0
	UART1
	UART2

#### NOTICE

- If the register C(n)\_AXI\_CONF[arprot] is 0x0, the read attribute of the channel is secure. In this case, the source address space can only be the secure DDR area or shared DDR area.
- If the register C(n)\_AXI\_CONF[arprot] is 0x2, the read attribute of the channel is non-secure. In this case, the source address space can only be the non-secure



DDR area or shared DDR area.

- If the register C(n)\_AXI\_CONF[awprot] is 0x0, the write attribute of the channel is secure. In this case, the source address space can only be the secure DDR area or shared DDR area.
- If the register C(n)\_AXI\_CONF[awprot] is 0x2, the write attribute of the channel is non-secure. In this case, the source address space can only be the non-secure DDR area or shared DDR area.
- If the access permission is incorrect, the DMAC reports the data transfer error interrupt of the channel. After the interrupt is cleared, the channel can be used.
- When a peripheral is accessed, the read and write security attributes are not distinguished.

### 3.6.3.3 Basic Transfers

The DMAC supports the following basic transfer modes: memory-to-memory, memory-to-peripherals, and peripherals-to-memory. Up to 65535 bytes of data can be transferred at a time.

- Memory-to-memory transfer: The source address and destination address are both physical memory addresses.
- Memory-to-peripherals transfer: The source address is the physical memory address, but the destination address is the TX FIFO register address of a peripheral. In addition, the destination address must be fixed during the transfer process.
- Peripherals-to-memory transfer: The source address is the RX FIFO register address of a peripheral, but the destination address is the physical memory address. In addition, the source address must be fixed during the transfer process.
- If the source address is not incremental, it must align with the transfer bit width of the source end. Otherwise, a configuration error interrupt is reported. If the source end is a peripheral, the size of the transfer bid width of the source end must be the same as the bid width of the RX FIFO peripheral.
- If the destination address is not incremental, it must align with the transfer bit width of the destination end. Otherwise, a configuration error interrupt is reported. If the destination end is a peripheral, the size of the transfer bid width of the destination end must be the same as the bid width of the TX FIFO peripheral.

#### NOTICE

- For the memory-to-peripherals and peripherals-to-memory transfers, when the data volume of a transfer is greater than that of a burst transfer, the DMAC responds to the burst request first, ignoring the single request.
- When the DMAC executes memory-to-peripherals transfers, the threshold of the



peripheral TX FIFO and the burst data amount of the DMA destination end may be inappropriate. As a result, the number of bursts in a transfer is greater than the remaining space of the TX FIFO, and data loss occurs.

- When the DMAC executes peripherals-to-memory transfers, the threshold of the peripheral RX FIFO and the burst data amount of the DMA source end may be inappropriate. As a result, the remaining data amount of the RX FIFO may be greater than the burst data amount of the DMA source end in a transfer but less than the RX threshold. In addition, the peripheral will not generate RX burst requests. The DMA only responds to RX burst requests. Therefore, the DMA is suspended.

### 3.6.3.4 Linked List Transfer

The DMAC supports the hardware linked list. That is, the DMA reads information of the next node from the linked list space preset by the CPU, and automatically starts the transfer of the next node. No CPU intervention is required. One linked list node can transfer a maximum of 65535 bytes of data.

The three basic transfers all support the DMA linked list function. To enable the linked list function, set the register [C\(n\)\\_LLI\\_L](#)[chain\_en] to 0x2. In addition, registers [C\(n\)\\_LLI\\_L](#)[chain\_en] in the linked list information of all nodes except the end node of the linked list must also be 0x2. For the end of the linked list, set the bit[chain\_en] of the [C\(n\)\\_LLI\\_L](#) register in the node to 0x0, indicating that the channel will be disabled automatically after all data blocks of the node are transferred.

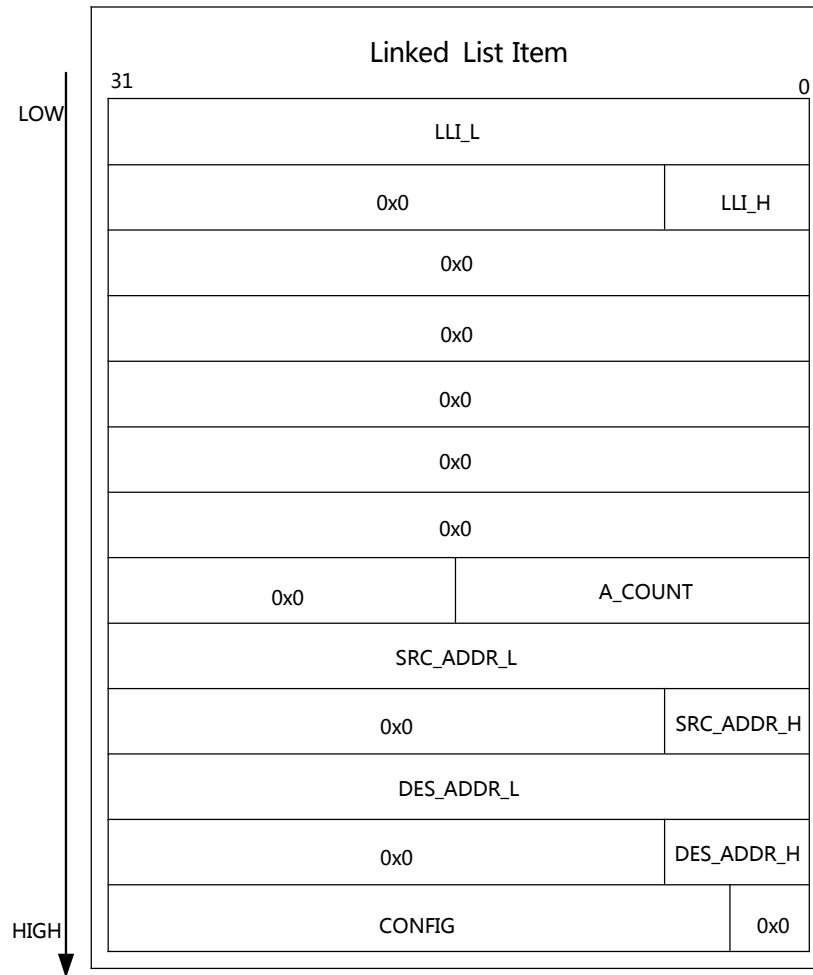
The CPU should configure the linked list address (registers [C\(n\)\\_LLI\\_L](#)[lli\_l] and [C\(n\)\\_LLI\\_H](#)[lli\_h]) first. After the configuration of the previous node is finished, the DMA automatically loads the information of the next preset linked list from the DDR space. Then, it automatically updates the linked list information to the following registers: [C\(n\)\\_SRC\\_ADDR\\_L](#), [C\(n\)\\_SRC\\_ADDR\\_H](#), [C\(n\)\\_DES\\_ADDR\\_L](#), [C\(n\)\\_DES\\_ADDR\\_H](#), [C\(n\)\\_LLI\\_L](#), [C\(n\)\\_LLI\\_H](#), [C\(n\)\\_CNT0](#), and [C\(n\)\\_CONFIG](#)[31:1].

Note that if the source address space of a channel is a secure memory, the linked list information must also be stored in the secure memory. Otherwise, the linked list information is stored in a non-secure memory.

#### NOTE

- If an error occurs in the configuration parameters updated by the linked list, the transfer is terminated directly, and a configuration error interrupt is generated.
- When a linked list read error or configuration error occurs and the linked list node interrupt of the previous node is enabled, an error interrupt as well as a linked list node completion interrupt are generated; otherwise, only an error interrupt is generated.
- The ch\_en value of the [C\(n\)\\_CONFIG](#) is not used by the internal DMAC.

Figure 3-5 shows the formats of the linked list stored in the DDR.

**Figure 3-5** Formats of the linked list stored in the DDR

### 3.6.3.5 Interrupts and Status

You can query the register **CH\_STAT** to determine whether the corresponding channel is working. The channel parameters cannot be configured if the channel is in the working state.

Channel completion interrupts are used to indicate the completion of transfers in the channel.

Linked list completion interrupts are used to indicate the completion of transfers in the linked list node. (The **C(n)\_CONFIG[1]** needs to be enabled.) Each node of linked lists can determine whether to report an interrupt based on linked list parameters. Therefore, linked lists can enable or disable the interrupt of each node. However, it should be noted that the last node of the linked list only generates transfer completion interrupts rather than node completion interrupts.

Configuration error interrupts are used to indicate invalid configurations of the channel or the linked list.



Linked list read error interrupts and data transfer error interrupts are used to indicate response errors of the bus. The former takes place when the linked list information is being read, and the latter takes place when data is being transferred.

The DMAC supports query of the current information of each channel. The register [C\(n\)\\_CURR\\_CNT0](#) is used to query the remaining data amount. The register [C\(n\)\\_CURR\\_SRC\\_ADDR\\_L](#) and [C\(n\)\\_CURR\\_SRC\\_ADDR\\_H](#) are used to query the source address of the current transfer. The register [C\(n\)\\_CURR\\_DES\\_ADDR\\_L](#) and [C\(n\)\\_CURR\\_DES\\_ADDR\\_H](#) are used to query the destination address of the current transfer.

## DNOTE

- You must clear all interrupts before data transfer. This ensures that interrupts are generated during running of the DMAC.
- If each node of the linked list transfers few data, some node interrupts cannot be queried and may be cleared because it takes the CPU some time to respond to node interrupts and clear interrupts.
- When channel transfer is completed normally or an error interrupt is generated, channels are automatically disabled.
- For a peripheral transfer, data should be queried when peripheral requests are invalid. The actual remaining data amount of the DMAC and the transfer address for the next peripheral request are queried.
- For a memory-to-memory transfer, the DMAC always transfers data. The queried data amount differs from the actually transferred data amount. Therefore, it is meaningless to query the register during memory-to-memory transfers.

### 3.6.3.6 Channel Security Configuration

You can configure the arprot and awprot fields of the register [C\(n\)\\_AXI\\_CONF](#) to implement the security attributes of each channel. The field configuration complies with the AXI bus protocol. That is, bit[6] and bit[18] of register [C\(n\)\\_AXI\\_CONF](#) reflect the read and write security attributes of channel n. If the security flag bit of the arprot or awprot field is 0x0, the channel is a secure channel. Otherwise, the channel is a non-secure channel.

The security attributes of all channels are secure by default.

#### NOTE

The **awprot** and **arprot** fields of the [C\(n\)\\_AXI\\_CONF](#) register can be configured only by the secure CPU.

## 3.6.4 Working Mode

### 3.6.4.1 Clock and Reset

You must enable the clock and perform reset before using the DMAC.

**Step 1** Write 0x1 to the CRG register [PERI\\_CRG2720](#) [edma\_axi\_cken] to enable the AXI clock gating of the DMAC module. Write 0x1 to the CRG register [PERI\\_CRG2720](#) [edma\_apb\_cken] to enable the APB clock gating of the DMAC module.



**Step 2** Write 0x1 to the CRG register[PERI\\_CRG2720](#)[edma\_srst\_req], and reset the DMAC module.

**Step 3** Write 0x0 to the CRG register[PERI\\_CRG2720](#)[edma\_srst\_req], and cancel resetting the DMAC module.

----End

### 3.6.4.2 Access Mode Setting

By default, all channels of the DMAC have secure attributes. In section[3.6.4.3 "Initialization"](#), section[3.6.4.4 "Basic Transfer"](#), section[3.6.4.8 "Linked List Transfer"](#), and section[3.6.4.9 "Interrupt Handling"](#), the access devices can only be secure processors. The accessed DDR memory can have secure attributes or shared attributes.

If a non-secure processor is required to access the DMAC, ensure that the following step is performed for the secure processor:

**Step 1** Set the arprot field (bit[7:5]) and awprot field (bit[19:17]) of the[C\(n\)\\_AXI\\_CONF](#) register to 0x010 for all channels to convert the secure attributes of channels to non-secure attributes.

**Step 2** In this case, all channels of the DMAC have non-secure attributes. In section[3.6.4.3 "Initialization"](#), section[3.6.4.4 "Basic Transfer"](#), section[3.6.4.8 "Linked List Transfer"](#), and section[3.6.4.9 "Interrupt Handling"](#), the access devices can only be non-secure processors. The accessed DDR memory can have non-secure attributes or shared attributes.

----End

### 3.6.4.3 Initialization

After the reset is deasserted and the access mode is set, the DMAC should be initialized. Perform the following steps:

**Step 1** Configure the priority to control the register[CH\\_PRI](#) based on the actual scenario.

**Step 2** Write 0x1 to all valid bits of the original interrupt registers including[INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#), and clear all interrupt status.

**Step 3** Configure interrupt mask registers including[INT\\_TC1\\_RAW](#), [INT\\_TC2\\_MASK](#), [INT\\_ERR1\\_MASK](#), [INT\\_ERR2\\_MASK](#), and [INT\\_ERR3\\_MASK](#) based on interrupt requirements.

**Step 4** Write 0x0 to the register[C\(n\)\\_CONFIG](#)[ch\_en] of each channel in sequence and disable the DMAC channel.

----End



### 3.6.4.4 Basic Transfer

After the DMAC is initialized, the DMAC can transmit data only when the channel is configured and enabled.

- Step 1** Read the register [INT\\_STAT](#) to search for idle channels and select a channel n.
- Step 2** Write  $0x1 << n$  (shifting 0x1 to the left by n bits) to the registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#), and clear the interrupt status of the selected channel.
- Step 3** Write data to the channel register [C\(n\)\\_SRC\\_ADDR\\_L](#) and [C\(n\)\\_SRC\\_ADDR\\_H](#) to set the start physical address of the source end.
- Step 4** Write data to the channel register [C\(n\)\\_DES\\_ADDR\\_L](#) and [C\(n\)\\_DES\\_ADDR\\_H](#) to set the start physical address of the destination end.
- Step 5** Write 0x0 to the channel register [C\(n\)\\_LLI\\_L](#)[chain\_en], indicating that the current transfer is a non-linked list transfer.
- Step 6** Write data to the channel register [C\(n\)\\_CNT0](#)[a\_count] to set the volume of data to be transferred. The maximum data volume is 65535 bytes.
- Step 7** Write data to the channel register [C\(n\)\\_CONFIG](#) to set the address type, transfer bit width, transfer burst length, transfer type and peripheral ID.
- Step 8** Write 0x1 to the channel register [C\(n\)\\_CONFIG](#)[ch\_en] to enable the channel. The DMAC starts to transmit data.
- Step 9** Enable the corresponding peripheral again if the data is transferred between memories and peripherals, and enable the DMA function. For details, see the I<sup>2</sup>C, UART, and SPI sections in chapter 12 "Peripherals."

----End

#### NOTE

- When the channel is busy, the DMAC does not respond to the operation request if 0x1 is written to [C\(n\)\\_CONFIG](#)[ch\_en].
- If a configuration error interrupt is generated after the channel is enabled, no data will be actually transferred.
- Registers can be configured only when the channel is idle. If the channel is working, register values will not change, and the DMAC does not notify software of the operation failure.
- Writing 1 (bitwise operation) to raw interrupt registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#) clears bit fields. Writing 0 to registers and reading data from registers does not change the original status.

### 3.6.4.5 Memory-to-Memory Transfer

An example procedure of a memory-to-memory transfer is as follows:

- Step 1** Read the register [CH\\_STAT](#), and detect that channel 0 is idle.



- Step 2** Write 0x1 to the registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#).
- Step 3** Write 0x84000000 to the registers C(0)\_SRC\_ADDR\_L and C(0)\_SRC\_ADDR\_H.
- Step 4** Write 0x85000000 to the registers C(0)\_DES\_ADDR\_L and C(0)\_DES\_ADDR\_H.
- Step 5** Write 0x0 to the register C(0)\_LLI\_L[1:0].
- Step 6** Write 0x1000 to the register C(0)\_CNT0.
- Step 7** Write 0xcff33000 to the register C(0)\_CONFIG.
- Step 8** Write 0x1 to the register C(0)\_CONFIG[ch\_en].

----End

### 3.6.4.6 Memory-to-Peripherals Transfer

An example procedure of a memory-to-I<sup>C</sup>0 transfer is as follows:

- Step 1** Read the register [CH\\_STAT](#), and detect that channel 1 is idle.
- Step 2** Write 0x2 to the registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#).
- Step 3** Write 0x84000000 to the registers C(1)\_SRC\_ADDR\_L and C(1)\_SRC\_ADDR\_H.
- Step 4** Write 0x11060000 to the registers C(1)\_DES\_ADDR\_L and C(1)\_DES\_ADDR\_H.
- Step 5** Write 0x0 to the register C(1)\_LLI\_L[1:0].
- Step 6** Write 0x10 to the register C(1)\_CNT0.
- Step 7** Write 0x80000014 to the register C(1)\_CONFIG.
- Step 8** Write 0x1 to the register C(1)\_CONFIG[ch\_en].
- Step 9** Complete the I<sup>C</sup>0 configuration and enable the DMA TX function.

----End

An example procedure of a memory-to-UART0 transfer is as follows:

- Step 1** Read the register [CH\\_STAT](#), and detect that channel 1 is idle.
- Step 2** Write 0x2 to the registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#).
- Step 3** Write 0x84000000 to the registers C(1)\_SRC\_ADDR\_L and C(1)\_SRC\_ADDR\_H.
- Step 4** Write 0x11040000 to the registers C(1)\_DES\_ADDR\_L and C(1)\_DES\_ADDR\_H.
- Step 5** Write 0x0 to the register C(1)\_LLI\_L[1:0].
- Step 6** Write 0x10 to the register C(1)\_CNT0.



- Step 7** Write 0x800000C4 to the register C(1)\_CONFIG.
- Step 8** Write 0x1 to the register C(1)\_CONFIG[ch\_en].
- Step 9** Complete the UART0 configuration and enable the DMA TX function.
- End

An example procedure of a memory-to-SSP0 transfer is as follows:

- Step 1** Read the register [CH\\_STAT](#), and detect that channel 1 is idle.
- Step 2** Write 0x2 to the registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#).
- Step 3** Write 0x84000000 to the registers C(1)\_SRC\_ADDR\_L and C(1)\_SRC\_ADDR\_H.
- Step 4** Write 0x11070008 to the registers C(1)\_DES\_ADDR\_L and C(1)\_DES\_ADDR\_H.
- Step 5** Write 0x0 to the register C(1)\_LLI\_L[1:0].
- Step 6** Write 0x10 to the register C(1)\_CNT0.
- Step 7** Write 0x80011074 to the register C(1)\_CONFIG.
- Step 8** Write 0x1 to the register C(1)\_CONFIG[ch\_en].
- Step 9** Complete the SSP0 configuration and enable the DMA TX function.

----End

### 3.6.4.7 Peripherals-to-Memory Transfer

An example procedure of an I<sup>2</sup>C0-to-memory transfer is as follows:

- Step 1** Read the register [CH\\_STAT](#), and detect that channel 2 is idle.
- Step 2** Write 0x4 to the registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#).
- Step 3** Write 0x11060000 to the registers C(2)\_SRC\_ADDR\_L and C(2)\_SRC\_ADDR\_H.
- Step 4** Write 0x85000000 to the registers C(2)\_DES\_ADDR\_L and C(2)\_DES\_ADDR\_H.
- Step 5** Write 0x0 to the register C(2)\_LLI\_L[1:0].
- Step 6** Write 0x10 to the register C(2)\_CNT0.
- Step 7** Write 0x40000004 to the register C(2)\_CONFIG.
- Step 8** Write 0x1 to the register C(2)\_CONFIG[ch\_en].
- Step 9** Complete the I<sup>2</sup>C0 configuration and enable the DMA RX function.

----End



An example procedure of a UART0-to-memory transfer is as follows:

- Step 1** Read the register [CH\\_STAT](#), and detect that channel 2 is idle.
- Step 2** Write 0x4 to the registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#).
- Step 3** Write 0x11040000 to the registers C(2)\_SRC\_ADDR\_L and C(2)\_SRC\_ADDR\_H.
- Step 4** Write 0x85000000 to the registers C(2)\_DES\_ADDR\_L and C(2)\_DES\_ADDR\_H.
- Step 5** Write 0x0 to the register C(2)\_LLI\_L[1:0].
- Step 6** Write 0x10 to the register C(2)\_CNT0.
- Step 7** Write 0x400000A8 to the register C(2)\_CONFIG.
- Step 8** Write 0x1 to the register C(2)\_CONFIG[ch\_en].
- Step 9** Complete the UART0 configuration and enable the DMA RX function.

----End

An example procedure of an SSP0-to-memory transfer is as follows:

- Step 1** Read the register [CH\\_STAT](#), and detect that channel 2 is idle.
- Step 2** Write 0x4 to the registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#).
- Step 3** Write 0x11070008 to the registers C(2)\_SRC\_ADDR\_L and C(2)\_SRC\_ADDR\_H.
- Step 4** Write 0x85000000 to the registers C(2)\_DES\_ADDR\_L and C(2)\_DES\_ADDR\_H.
- Step 5** Write 0x0 to the register C(2)\_LLI\_L[1:0].
- Step 6** Write 0x10 to the register C(2)\_CNT0.
- Step 7** Write 0x40011068 to the register C(2)\_CONFIG.
- Step 8** Write 0x1 to the register C(2)\_CONFIG[ch\_en].
- Step 9** Complete the SSP0 configuration and enable the DMA RX function.

----End

### 3.6.4.8 Linked List Transfer

Read the register [INT\\_STAT](#) to search for idle channels and select a channel n.

- Step 1** Write 0x1<<n to registers [INT\\_TC1\\_RAW](#), [INT\\_TC2\\_RAW](#), [INT\\_ERR1\\_RAW](#), [INT\\_ERR2\\_RAW](#), and [INT\\_ERR3\\_RAW](#) to clear the interrupt status of selected channels.
- Step 2** Software allocates linked list space. The address of the linked list space must be 512-bit-aligned. Fill the linked list space with the linked list content in sequence. For details, see section "[3.6.3.4 Linked List Transfer](#)".



- Step 3** Write data to the channel register `C(n)_SRC_ADDR_L` and `C(n)_SRC_ADDR_H` to set the start physical address of the source end of the first linked list node.
- Step 4** Write data to the channel register `C(n)_DES_ADDR_L` and `C(n)_DES_ADDR_H` to set the start physical address of the destination end of the first linked list node.
- Step 5** Write the channel registers `C(n)_LLI_L[lli_l]` and `C(n)_LLI_H[lli_h]`, and set the start address of the linked list space.
- Step 6** Write the channel register `C(n)_LLI_L[chain_en]` to 0x2, indicating that the node is not the end of the linked list.
- Step 7** Write data to the channel register `C(n)_CNT0[a_count]` to set the volume of data to be transferred in this node. The maximum data volume is 65535 bytes.
- Step 8** Write the channel register `C(n)_CONFIG`, and set the address type, transfer bit width, transfer burst length, transfer type, peripheral ID, and linked list interrupt enable of the node.
- Step 9** Write the channel register `C(n)_CONFIG[ch_en]` to 0x1. The DMAC starts to transmit data.
- Step 10** Enable the corresponding peripheral again if the data is transferred between memories and peripherals, and enable the DMA function. For details, see the I2C, UART, and SPI sections in chapter 12 "Peripherals."

----End

#### NOTE

For details, see the description in section "[3.6.4.4 Basic Transfer](#)."

### 3.6.4.9 Interrupt Handling

The procedure of handling interrupts is as follows:

- Step 1** Read the interrupt status register `INT_STAT` to search for the channel that sends an interrupt request. If interrupt requests are reported in multiple channels at the same time, the channel of the highest priority is preferred.
- Step 2** Read the register `INT_TC1` to query whether the value of the selected bit is 0x1 and determine whether the interrupt sent by the corresponding channel is a transfer completion interrupt. If the value is 0x1, the interrupt is a transfer completion interrupt. In this case, go to Step 4; otherwise, go to Step 3.
- Step 3** Read the registers `INT_ERR1`, `INT_ERR2`, and `INT_ERR3` to query whether the value of the selected bit is 0x1 and determine whether the interrupt sent by the corresponding channel is an error interrupt. If the value is 0x1, the interrupt is a transfer completion interrupt. In this case, go to Step 5; otherwise, go to Step 6.
- Step 4** Handle the transfer completion interrupt. The specific procedure is as follows:
1. Write 0x1 to the selected bit of `INT_TC1_RAW` to clear the interrupt status of the corresponding channel.



2. Remove or use up the data in the memory. If necessary, configure and enable the channel again.
3. End interrupt operations.

**Step 5** Handle the error interrupt. The specific procedure is as follows:

1. Clear the error interrupt status based on the error type. If the error occurs in the configuration, write 0x1 to the selected bit of [INT\\_ERR1\\_RAW](#). If the error occurs in the transfer, write 0x1 to the selected bit of [INT\\_ERR2\\_RAW](#). If the error occurs in the linked list transfer, write 0x1 to the selected bit of [INT\\_ERR3\\_RAW](#).
2. Provide the error information. If necessary, configure and enable the channel again.
3. End interrupt operations.

**Step 6** Handle the transfer completion interrupt of a linked list node. The specific procedure is as follows:

1. Write 0x1 to the selected bit of [INT\\_TC2\\_RAW](#) to clear the interrupt status of the corresponding channel.
2. Remove or use up the data in the memory.
3. End interrupt operations.

----End

### 3.6.5 Register Summary

[Table 3-11](#) describes the value ranges and meanings of the variable in the offset addresses of registers.

**Table 3-11** Variable in the register offset addresses

Variable	Value Range	Description
n	0-3	Number of channels supported

[Table 3-12](#) describes the DMAC registers.

**Table 3-12** DMAC registers (base address: 0x1028\_0000)

Offset Address	Register	Description	Page Number
0x0000	INT_STAT	Interrupt status register	<a href="#">3-95</a>
0x0004	INT_TC1	Channel transfer completion interrupt status register	<a href="#">3-95</a>



Offset Address	Register	Description	Page Number
0x0008	INT_TC2	Linked list node transfer completion interrupt status register	<a href="#">3-95</a>
0x000C	INT_ERR1	Configuration error interrupt status register	<a href="#">3-96</a>
0x0010	INT_ERR2	Data transfer error interrupt status register	<a href="#">3-96</a>
0x0014	INT_ERR3	Linked list read error interrupt status register	<a href="#">3-97</a>
0x0018	INT_TC1_MASK	Channel transfer completion interrupt mask register	<a href="#">3-97</a>
0x001C	INT_TC2_MASK	Linked list node transfer completion interrupt mask register	<a href="#">3-97</a>
0x0020	INT_ERR1_MASK	Configuration error interrupt mask register	<a href="#">3-98</a>
0x0024	INT_ERR2_MASK	Data transfer error interrupt mask register	<a href="#">3-98</a>
0x0028	INT_ERR3_MASK	Linked list read error interrupt mask register	<a href="#">3-98</a>
0x0404 + 0x20 × n	C(n)_CURR_CNT0	Remaining data amount register of channel n	<a href="#">3-99</a>
0x0408 + 0x20 × n	C(n)_CURR_SRC_A DDR_L	Source address lower-bit register of channel n	<a href="#">3-99</a>
0x040C + 0x20 × n	C(n)_CURR_SRC_A DDR_H	Source address upper-bit register of channel n	<a href="#">3-99</a>
0x0410 + 0x20 × n	C(n)_CURR_DEST ADDR_L	Destination address lower-bit register of channel n	<a href="#">3-100</a>
0x0414 + 0x20 × n	C(n)_CURR_DEST ADDR_H	Destination address upper-bit register of channel n	<a href="#">3-100</a>
0x0600	INT_TC1_RAW	Raw channel transfer completion interrupt status register	<a href="#">3-100</a>
0x0608	INT_TC2_RAW	Raw linked list node transfer completion interrupt status register	<a href="#">3-101</a>



Offset Address	Register	Description	Page Number
0x0610	INT_ERR1_RAW	Raw configuration error interrupt status register	<a href="#">3-101</a>
0x0618	INT_ERR2_RAW	Raw data transfer error interrupt status register	<a href="#">3-102</a>
0x0620	INT_ERR3_RAW	Raw linked list read error interrupt status register	<a href="#">3-102</a>
0x0688	CH_PRI	Priority control register	<a href="#">3-103</a>
0x0690	CH_STAT	DMA status register	<a href="#">3-104</a>
0x0694	SEC_CTRL	Global security control register	<a href="#">3-104</a>
0x0800 + 0x40 × n	C(n)_LLI_L	Linked list address lower-bit configuration register of channel n	<a href="#">3-105</a>
0x0804 + 0x40 × n	C(n)_LLI_H	Linked list address upper-bit configuration register of channel n	<a href="#">3-105</a>
0x081C + 0x40 × n	C(n)_CNT0	Transfer length configuration register of channel cn	<a href="#">3-106</a>
0x0820 + 0x40 × n	C(n)_SRC_ADDR_L	Source address lower-bit configuration register of channel n	<a href="#">3-106</a>
0x0824 + 0x40 × n	C(n)_SRC_ADDR_H	Source address upper-bit configuration register of channel n	<a href="#">3-106</a>
0x0828 + 0x40 × n	C(n)_DES_ADDR_L	Destination address lower-bit configuration register of channel n	<a href="#">3-107</a>
0x082C + 0x40 × n	C(n)_DES_ADDR_H	Destination address upper-bit configuration register of channel n	<a href="#">3-107</a>
0x0830 + 0x40 × n	C(n)_CONFIG	Configuration register of channel n	<a href="#">3-107</a>
0x0834 + 0x40 × n	C(n)_AXI_CONF	AXI special operation configuration register of channel n	<a href="#">3-110</a>
0x0f00	DMAC_VERSION	Version register	<a href="#">3-111</a>



### 3.6.6 Register Description

#### INT\_STAT

INT\_STAT is an interrupt status register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RO	int_stat	Status of the masked interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: No interrupt is generated. 1: An interrupt is generated. The interrupt request may be an error interrupt or a transfer completion interrupt.	0x0

#### INT\_TC1

INT\_TC1 is a channel transfer completion interrupt status register.

Offset Address: 0x0004 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RO	int_tc1	Status of the masked channel transfer completion interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: No interrupt is generated. 1: An interrupt is generated.	0x0

#### INT\_TC2

INT\_TC2 is a linked list node transfer completion interrupt status register.

Offset Address: 0x0008 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000



Bits	Access	Name	Description	Reset
[3:0]	RO	int_tc2	Status of the masked linked list node transfer completion interrupts for each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## INT\_ERR1

INT\_ERR1 is a configuration error interrupt status register.

Offset Address: 0x000C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RO	int_err1	Status of the masked configuration error interrupts for each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## INT\_ERR2

INT\_ERR2 is a data transfer error interrupt status register.

Offset Address: 0x0010 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RO	int_err2	Status of the masked data transfer error interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: No interrupt is generated. 1: An interrupt is generated.	0x0



## INT\_ERR3

INT\_ERR3 is a linked list read error interrupt status register.

Offset Address: 0x0014 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RO	int_err3	Status of the masked linked list read error interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## INT\_TC1\_MASK

INT\_TC1\_MASK is a channel transfer completion interrupt mask register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW	int_tc1_mask	Mask status of the transfer completion interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: The interrupt is masked. 1: The interrupt is not masked.	0x0

## INT\_TC2\_MASK

INT\_TC2\_MASK is a linked list node transfer completion interrupt mask register.

Offset Address: 0x001C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW	int_tc2_mask	Mask status of the linked list node transfer completion interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0.	0x0



Bits	Access	Name	Description	Reset
			0: The interrupt is masked. 1: The interrupt is not masked.	

## INT\_ERR1\_MASK

INT\_ERR1\_MASK is a configuration error interrupt mask register.

Offset Address: 0x0020 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW	int_err1_mask	Mask status of configuration error interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: The interrupt is masked. 1: The interrupt is not masked.	0x0

## INT\_ERR2\_MASK

INT\_ERR2\_MASK is a data transfer error interrupt mask register.

Offset Address: 0x0024 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW	int_err2_mask	Mask status of data transfer error interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: The interrupt is masked. 1: The interrupt is not masked.	0x0

## INT\_ERR3\_MASK

INT\_ERR3\_MASK is a linked list read error interrupt mask register.

Offset Address: 0x0028 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW	int_err3_mask	Mask status of linked list read error interrupts of each DMAC channel. Bit[3:0] corresponds to channels 3–0. 0: The interrupt is masked. 1: The interrupt is not masked.	0x0

## C(n)\_CURR\_CNT0

C(n)\_CURR\_CNT0 is a remaining data amount register of channel n.

Offset Address: 0x0404 + 0x20 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RO	curr_a_count	Number of remaining bytes in the current node of channel n	0x0000

## C(n)\_CURR\_SRC\_ADDR\_L

C(n)\_CURR\_SRC\_ADDR\_L is a source address lower-bit register of channel n. This register inherits the security attributes of the channel.

Offset Address: 0x0408 + 0x20 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	curr_src_addr_l	Lower 32 bits of the source address that is being transferred by channel n	0x00000000

## C(n)\_CURR\_SRC\_ADDR\_H

C(n)\_CURR\_SRC\_ADDR\_H is a source address upper-bit register of channel n.

Offset Address: 0x040C + 0x20 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000



[1:0]	RO	curr_src_addr_h	Upper 2 bits of the source address that is being transferred by channel n	0x0
-------	----	-----------------	---	-----

### C(n)\_CURR\_DES\_ADDR\_L

C(n)\_CURR\_DES\_ADDR\_L is a destination address lower-bit register of channel n. This register inherits the security attributes of the channel.

Offset Address: 0x0410+0x20 x n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	curr_des_addr_l	Lower 32 bits of the destination address that is being transferred by channel n	0x00000000

### C(n)\_CURR\_DES\_ADDR\_H

C(n)\_CURR\_DES\_ADDR\_H is a destination address upper-bit register of channel n.

Offset Address: 0x0414+0x20 x n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1:0]	RO	curr_des_addr_h	Upper 2 bits of the destination address that is being transferred by channel n	0x0

### INT\_TC1\_RAW

INT\_TC1\_RAW is a raw channel transfer completion interrupt status register. Each bit field of this register inherits the security attributes of the corresponding channel.

Offset Address: 0x0600 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RWC	int_tc1_raw	Raw status of channel transfer completion interrupts of each channel. Bit[3:0] corresponds to channels 3–0. Read the register: 0: No interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
			1: An interrupt is generated. Write the register: 0: Reserve the previous values. 1: Clear channel transfer completion interrupts.	

## INT\_TC2\_RAW

INT\_TC2\_RAW is a raw linked list node transfer completion interrupt status register. Each bit field of this register inherits the security attributes of the corresponding channel.

Offset Address: 0x0608 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RWC	int_tc2_raw	Raw status of linked list node transfer completion interrupts for each channel. Bit[3:0] corresponds to channels 3–0. Read the register: 0: No interrupt is generated. 1: An interrupt is generated. Write the register: 0: Reserve the previous values. 1: Clear linked list node transfer completion interrupts.	0x0

## INT\_ERR1\_RAW

INT\_ERR1\_RAW is a raw configuration error interrupt status register. Each bit field of this register inherits the security attributes of the corresponding channel.

Offset Address: 0x0610 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RWC	int_err1_raw	Raw status of the configuration error interrupts for each channel. Bit[3:0]	0x0



Bits	Access	Name	Description	Reset
			corresponds to channels 3–0.  Read the register: 0: No interrupt is generated. 1: An interrupt is generated.  Write the register: 0: Reserve the previous values. 1: Clear configuration error interrupts.	

## INT\_ERR2\_RAW

INT\_ERR2\_RAW is a raw data transfer error interrupt status register. Each bit field of this register inherits the security attributes of the corresponding channel.

Offset Address: 0x0618 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RWC	int_err2_raw	Raw status of data transfer error interrupts for each channel. Bit[3:0] corresponds to channels 3–0.  Read the register: 0: No interrupt is generated. 1: An interrupt is generated.  Write the register: 0: Reserve the previous values. 1: Clear data transfer error interrupts.	0x0

## INT\_ERR3\_RAW

INT\_ERR3\_RAW is a raw linked list read error interrupt status register. Each bit field of this register inherits the security attributes of the corresponding channel.

Offset Address: 0x0620 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RWC	int_err3_raw	Raw status of linked list read error interrupts for each channel. Bit[3:0]	0x0



Bits	Access	Name	Description	Reset
			corresponds to channels 3–0. Read the register: 0: No interrupt is generated. 1: An interrupt is generated. Write the register: 0: Reserve the previous values. 1: Clear linked list read error interrupts.	

## CH\_PRI

CH\_PRI is a priority control register, which can be accessed only by the secure CPUs.

Offset Address: 0x0688 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:6]	RW	ch3_pri	Priority configuration register of channel 3 00: priority 0 (highest) 01: priority 1 10: priority 2 11: priority 3 (lowest)	0x0
[5:4]	RW	ch2_pri	Priority configuration register of channel 2 00: priority 0 (highest) 01: priority 1 10: priority 2 11: priority 3 (lowest)	0x0
[3:2]	RW	ch1_pri	Priority configuration register of channel 1 00: priority 0 (highest) 01: priority 1 10: priority 2 11: priority 3 (lowest)	0x0
[1:0]	RW	ch0_pri	Priority configuration register of channel 0	0x0



Bits	Access	Name	Description	Reset
			00: priority 0 (highest) 01: priority 1 10: priority 2 11: priority 3 (lowest)	

## CH\_STAT

CH\_STAT is a DMA status register.

Offset Address: 0x0690 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RO	ch_stat	DMAC channel status register. Bit[3:0] corresponds to channels 3–0. 0: The corresponding DMAC channels are not operating. 1: The corresponding DMAC channels are operating.	0x0

## SEC\_CTRL

SEC\_CTRL is a DMA global security control register.

Offset Address: 0x0694 Total Reset Value: 0x0000\_0002

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x00000000
[1]	RW	intr_sec_ctrl	Interrupt security control register (accessed by only secure operations) 0: During secure operations, the interrupt information about only secure channels can be accessed. Secure interrupts are reported by using DMAC, and non-secure interrupts are reported by using DMAC_NS. 1: During secure operations, information about all channels can be accessed. Secure interrupts and non-secure	0x1



Bits	Access	Name	Description	Reset
			interrupts are reported by using DMAC. However, DMAC_NS reports only non-secure interrupts.	
[0]	RW	global_sec	Security control register for global registers (accessed by only secure operations) 0: <a href="#">CH_PRI</a> can be accessed only through secure operations. 1: <a href="#">CH_PRI</a> can be accessed through both secure and non-secure operations.	0x0

## C(n)\_LLI\_L

C(n)\_LLI\_L is a linked list address lower-bit configuration register of channel n. This register inherits the security attributes of the channel.

Offset Address: 0x0800 + 0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	RW	lli_l	Bit[31:6] of the next linked list node address of channel n. The linked list address must be 512-bit aligned.	0x00000000
[5:2]	-	reserved	Reserved	0x0
[1:0]	RW	chain_en	Channel link enable 00: The linked list link enable is invalid. 01: reserved 10: The linked list link enable is valid. 11: reserved	0x0

## C(n)\_LLI\_H

C(n)\_LLI\_H is a linked list address upper-bit configuration register of channel n.

Offset Address: 0x0804 + 0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1:0]	RW	lli_h	Bit[33:32] of the next linked list node	0x0



			address of channel n. The linked list address must be 512-bit aligned.	
--	--	--	--	--

## C(n)\_CNT0

C(n)\_CNT0 is a transfer length configuration register of channel n. This register inherits the security attributes of the channel.

Offset Address: 0x081C + 0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	a_count	Transfer length, that is, the transfer length of a single node of the linked list mode or the total transfer length of the non-linked list mode  The valid length is 1–65535 bytes.  A configuration error interrupt is reported when a_count is set to 0.	0x0000

## C(n)\_SRC\_ADDR\_L

C(n)\_SRC\_ADDR\_L is a source address lower-bit configuration register of channel n. This register inherits the security attributes of the channel.

Offset Address: 0x0820+0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	src_addr	Source address of channel n	0x00000000

## C(n)\_SRC\_ADDR\_H

C(n)\_SRC\_ADDR\_H is a source address upper-bit configuration register of channel n.

Offset Address: 0x0824+0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1:0]	RW	src_addr_h	Upper 2 bits of the source address of channel n	0x0



### C(n)\_DES\_ADDR\_L

C(n)\_DES\_ADDR\_L is a destination address lower-bit configuration register of channel n. This register inherits the security attributes of the channel.

Offset Address: 0x0828+0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	des_addr_l	Lower 32 bits of the destination address of channel n	0x00000000

### C(n)\_DES\_ADDR\_H

C(n)\_DES\_ADDR\_H is a destination address upper-bit configuration register of channel n.

Offset Address: 0x082C + 0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1:0]	RW	des_addr_h	Upper 2 bits of the destination address of channel n	0x0

### C(n)\_CONFIG

C(n)\_CONFIG is a configuration register of channel n. This register inherits the security attributes of the channel.

Offset Address: 0x0830 + 0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31]	RW	si	Source address increment 0: The source address is not incremented. 1: The source address is incremented once after a data segment is transferred. If the source device is a peripheral, the source address is not incremented. If the source device is a memory, the source address is incremented.	0x0
[30]	RW	di	Destination address increment	0x0



Bits	Access	Name	Description	Reset
			0: The destination address is not incremented. 1: The destination address is incremented once after a data segment is transferred. If the destination device is a peripheral, the destination address is not incremented. If the destination device is a memory, the destination address is incremented.	
[29:28]	-	reserved	Reserved	0x0
[27:24]	RW	sl	Burst length on the source side 0x0: 1 0x1: 2 0x2: 3 0x3: 4 0x4: 5 0x5: 6 0x6: 7 0x7: 8 0x8: 9 0x9: 10 0xA: 11 0xB: 12 0xC: 13 0xD: 14 0xE: 15 0xF: 16	0x0
[23:20]	RW	dl	Burst length on the destination side 0x0: 1 0x1: 2 0x2: 3 0x3: 4 0x4: 5 0x5: 6 0x6: 7 0x7: 8	0x0



Bits	Access	Name	Description	Reset
			0x8: 9 0x9: 10 0xA: 11 0xB: 12 0xC: 13 0xD: 14 0xE: 15 0xF: 16	
[19]	-	reserved	Reserved	0x0
[18:16]	RW	sw	Source data width 000: 8 bits 001: 16 bits 010: 32 bits 011: 64 bits 1xx: reserved	0x0
[15]	-	reserved	Reserved	0x0
[14:12]	RW	dw	Destination data width 000: 8 bits 001: 16 bits 010: 32 bits 011: 64 bits 1xx: reserved	0x0
[11:9]	-	reserved	Reserved	0x0
[8:4]	RW	peri	Reserved	0x00
[3:2]	RW	flow_ctrl	Peripheral request signal If the data is transferred from a memory to another one, these bits are ignored.	0x0
[1]	RW	itc_en	Flow control type and transfer type 00: Transfer between two memories, DMA flow control 01: Transfer between a memory and a peripheral, DMA flow control 10: Transfer between a memory and a peripheral, peripheral flow control 11: reserved	0x0



Bits	Access	Name	Description	Reset
[0]	RW	ch_en	<p>Linked list node transfer completion interrupt enable. This bit is used to configure whether to trigger a linked list node transfer completion interrupt after the current linked list node transfer is complete.</p> <p>0: A linked list node transfer completion interrupt is not triggered after the current linked list node transfer is complete.</p> <p>1: A linked list transfer completion interrupt is triggered after the current linked list node transfer is complete, but a linked list node transfer completion interrupt will not be triggered after the last linked list node transfer is complete.</p> <p>Note: A channel transfer completion interrupt is reported when all linked list node transfers are complete no matter whether this bit is set to 0 or 1.</p>	0x0

## C(n)\_AXI\_CONF

C(n)\_AXI\_CONF is an AXI special operation configuration register of channel n. This register inherits the security attributes of the channel. The awprot and arprot bit fields can be accessed only by secure CPUs.

Offset Address: 0x0834 + 0x40 × n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:20]	RW	awcache	Cache type of the AXI bus write command, complying with the AXI bus protocol	0x0
[19:17]	RW	awprot	Protection type of the AXI bus write command, complying with the AXI bus protocol	0x0
[16:12]	-	reserved	Reserved	0x00
[11:8]	RW	arcache	Cache type of the AXI bus read command, complying with the AXI bus protocol	0x0
[7:5]	RW	arprot	Protection type of the AXI bus read	0x0



Bits	Access	Name	Description	Reset
			command, complying with the AXI bus protocol	
[4:0]	-	reserved	Reserved	0x00

## DMAC\_VERSION

DATE\_VERSION is a version register.

Offset Address: 0x0f00 Total Reset Value: 0x0000\_C310

Bits	Access	Name	Description	Reset
[31:0]	RO	dmac_version	Controller version information	0x0000C310

## 3.7 Timer

### 3.7.1 Overview

The timer module implements the timing and counting functions. It not only serves as the system clock of the operating system, but also can be used by applications for timing and counting. The system provides 8 timers, four of which support only access by the secure OS.

### 3.7.2 Features

Timer has the following features:

- Provides 16-bit or 32-bit down counter that has a programmable 8-bit prescaler.
- The counting clock of the timer can be a 3 MHz clock.
- Supports three count modes: free-running mode, periodic mode, and one-shot mode.
- Loads the initial value through either of the following registers:[TIMERx\\_LOAD](#) and [TIMERx\\_BGLOAD](#).
- Reads the current count value at any time.
- Generates an interrupt when the count value is decreased to 0.



### 3.7.3 Function Description

The timer is a 32-bit or 16-bit configurable down counter. The counter value is decremented by 1 on each rising edge of the count clock. When the count value reaches 0, the timer generates an interrupt.

The timer supports three count modes:

- Free-running mode

The timer counts continuously. When the count value reaches 0, the timer wraps its value around to the maximum value automatically and then continues to count. When the count length is 32 bits, the maximum value is 0xFFFF\_FFFF. When the count length is 16 bits, the maximum value is 0xFFFF. In free-running mode, the count value is decremented immediately from the loaded value. When the value reaches 0, the value is wrapped around to the maximum value.

- Periodic mode

The timer counts continuously. When the count value reaches 0, the timer loads an initial value from [TIMERx\\_BGLOAD](#) again and then continues to count.

- One-shot mode

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

Each timer has a prescaler that divides the frequency of the working clock of each timer by 1, 16, or 256. In this way, flexible frequencies of the count clock are provided. An initial value is loaded to the timer as follows:

- An initial value can be loaded by writing [TIMERx\\_LOAD](#). When the timer works, if a value is written to [TIMERx\\_LOAD](#), the timer recounts starting from this value immediately. This method is applicable to all count modes.
- The count cycle in periodic mode can be set by writing [TIMERx\\_BGLOAD](#). The current count value of the timer is not affected immediately when [TIMERx\\_BGLOAD](#) is written. Instead, the timer continues to count until the count value reaches 0. Then the timer loads the new value of [TIMERx\\_BGLOAD](#) and starts to count.

### 3.7.4 Operating Mode

#### Initialization

The timer must be initialized when the system is initialized. To initialize timerX (X ranges from 0 to 11), do as follows:

**Step 1** Write to [TIMERx\\_LOAD](#) to load an initial value to the timer.



**Step 2** When the timer is required to work in periodic mode and the count cycle is different from the initial value loaded to the timer, write to [TIMERx\\_BGLOAD](#) to set the count cycle of the timer.

**Step 3** Write to [TIMERx\\_CONTROL](#) to set the count mode, counter length, prescaling factor, and interrupt mask of the timer, and then enable the timer to count.

----End

## Interrupt Processing

The timer is used to generate interrupts periodically. Therefore, interrupt processing is a process of activating and waiting the timing interrupt. To process an interrupt, do as follows:

**Step 1** Configure [TIMERx\\_INTCLR](#) to clear the interrupt of the timer.

**Step 2** Activate the processes of waiting for the interrupt and execute the process.

**Step 3** When all the processes of waiting for the interrupt are complete or the wait interrupt is in hibernate state, resume the interrupt and continue to execute the interrupted program.

----End

## 3.7.5 Register Summary

Each timer register has the same features except that their base addresses are different. The details about their base addresses are as follows:

- The base address of timer 0 is 0x1100\_0000
- The base address of timer 1 is 0x1100\_0020
- The base address of timer 2 is 0x1100\_1000
- The base address of timer 3 is 0x1100\_1020
- The base address of timer 4 (Sec\_Timer0) is 0x1100\_2000
- The base address of timer 5 (Sec\_Timer1) is 0x1100\_2020
- The base address of timer 6 (Sec\_Timer2) is 0x1100\_3000
- The base address of timer 7 (Sec\_Timer3) is 0x1100\_3020

### NOTE

The value range of X in timerX is 0–7. The registers of all controllers are the same. In this section, timer 0 registers are used for examples. Timers 4 to 7 can be configured to be accessed only by the secure system.

**Table 3-13** Summary of timer registers

Offset Address	Register	Description	Page Number
0x000	TIMERx_LOAD	Initial count value register	<a href="#">3-114</a>
0x004	TIMERx_VALUE	Current count value register	<a href="#">3-115</a>
0x008	TIMERx_CONTROL	Control register	<a href="#">3-115</a>
0x00C	TIMERx_INTCLR	Interrupt clear register	<a href="#">3-116</a>
0x010	TIMERx_RIS	Raw interrupt status register	<a href="#">3-117</a>
0x014	TIMERx_MIS	Masked interrupt status register	<a href="#">3-117</a>
0x018	TIMERx_BGLOAD	Initial count value register in periodic mode	<a href="#">3-117</a>

### 3.7.6 Register Description

#### **TIMERx\_LOAD**

TIMERx\_LOAD is an initial count value register. It is used to set the initial count value of each timer.

##### DNOTE

- The minimum valid value written to [TIMERx\\_LOAD](#) is 1.
- When the value 0 is written to [TIMERx\\_LOAD](#), the dual-timer module generates an interrupt immediately.

The differences between [TIMERx\\_LOAD](#) and [TIMERx\\_BGLOAD](#) are as follows:

If values are written to [TIMERx\\_BGLOAD](#) and [TIMERx\\_LOAD](#), the current count value on the next rising edge of TIMCLK is changed to the written value of [TIMERx\\_LOAD](#). As the value of [TIMERx\\_BGLOAD](#) changes when data is written to [TIMERx\\_LOAD](#), the value returned after [TIMERx\\_BGLOAD](#) read is the latest value that is written to [TIMERx\\_LOAD](#) and [TIMERx\\_BGLOAD](#). When the timer works in periodic mode and the count value decreases to 0, the initial value is loaded from [TIMERx\\_BGLOAD](#) to continue counting.

Offset Address: 0x000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer0_load	Initial count value of timer 0	0x00000000



## TIMERx\_VALUE

TIMERx\_VALUE is a current count value register. It shows the current value of the counter that is decremented.

After a value is written to [TIMERx\\_LOAD](#), [TIMERx\\_VALUE](#) immediately shows the newly loaded value of the counter in the PCLK domain without waiting for the clock edge of TIMCLK enabled by TIMCLKENx.

### NOTE

When a timer is in 16-bit mode, the 16 upper bits of the 32-bit [TIMERx\\_VALUE](#) are not set to 0 automatically. If the timer is switched from 32-bit mode to 16-bit mode and no data is written to [TIMERx\\_LOAD](#), the upper 16 bits of [TIMERx\\_VALUE](#) may be non-zero.

Offset Address: 0x004 Total Reset Value: 0xFFFF\_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RO	timer0_value	Current count value of timer 0 that is decremented	0xFFFFFFFF

## TIMERx\_CONTROL

TIMERx\_CONTROL is a control register. Each timer (timers 0-5) has one such register.

### DNOTE

When the periodic mode is selected, TIMERx\_CONTROL[timermode] must be set to 1 and TIMERx\_CONTROL[oneshot] must be set to 0.

Offset Address: 0x008 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7]	RW	timeren	Timer enable 0: disabled 1: enabled	0x0
[6]	RW	timermode	Timer count mode 0: free-running mode 1: periodic mode	0x0
[5]	RW	intenable	TIMERx_RIS interrupt mask 0: masked 1: not masked	0x0



Bits	Access	Name	Description	Reset
[4]	-	reserved	Reserved	0x0
[3:2]	RW	timerpre	Prescaling factor configuration 00: no prescaling. That is, the clock frequency of the timer is divided by 1. 01: 4-level prescaling. That is, the clock frequency of the timer is divided by 16. 10: 8-level prescaling. That is, the clock frequency of the timer is divided by 256. 11: undefined. If the bits are set to 11, 8-level prescaling is considered. That is, the clock frequency of the timer is divided by 256.	0x0
[1]	RW	timersize	Counter select 0: 16-bit counter 1: 32-bit counter	0x0
[0]	RW	oneshot	Count mode select 0: periodic mode or free-running mode 1: one-shot mode	0x0

## TIMERx\_INTCLR

TIMERx\_INTCLR is the interrupt clear register. The interrupt status of a counter is cleared after any operation is performed on this register. Each timer (timers 0-5) has one such register.

### NOTICE

This register is a write-only register. The timer clears interrupts when any value is written to this register. In addition, no value is recorded in this register and no default reset value is defined.

Offset Address: 0x00C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	WO	timer0_intclr	Writing this register clears the output interrupt of timer 0.	0x00000000



## **TIMERx\_RIS**

TIMERx\_RIS is a raw interrupt status register. Each timer (timers 0-5) has one such register.

Offset Address: 0x010 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:1]	RO	reserved	Reserved. Writing this register has no effect and reading this register returns 0.	0x00000000
[0]	RO	timer0ris	Raw interrupt status of timer 0 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## **TIMERx\_MIS**

TIMERx\_MIS is a masked interrupt status register.

Offset Address: 0x014 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer0mis	Masked interrupt status of timer 0 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

## **TIMERx\_BGLOAD**

TIMERx\_BGLOAD is an initial count value register in periodic mode. Each timer (timers 0-5) has one such register.

The TIMERx\_BGLOAD register contains the initial count value of the timer. This register is used to reload an initial count value when the count value of the timer reaches 0 in periodic mode.

In addition, this register provides another method of accessing [TIMERx\\_LOAD](#). The difference is that after a value is written to TIMERx\_BGLOAD, the timer does not count starting from the input value immediately.

Offset Address: 0x018 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	timer0bgload	Initial count value of timer 0 Note: This register differs from <a href="#">TIMERx_LOAD</a> . For details, see the descriptions of <a href="#">TIMERx_LOAD</a> .	0x00000000

## 3.8 Watchdog

### 3.8.1 Overview

The watchdog is used to transmit a reset signal to reset the entire system within a period after an exception occurs in the system. The system provides one watchdog module.

### 3.8.2 Features

The watchdog has the following features:

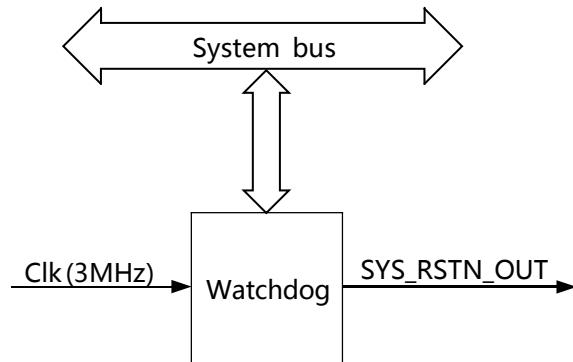
- Provides a 32-bit internal down counter.
- Supports the configurable timeout interval, namely, initial count value.
- Locks registers to avoid any modification to them.
- Supports the generation of timeout interrupts.
- Supports the generation of reset signals.
- Supports the debugging mode.

### 3.8.3 Function Description

#### Application Block Diagram

The system configures the parameter values of watchdog registers by using the system bus. The watchdog transmits interrupt requests to the system periodically. When the system does not respond to the interrupt requests (such as the suspend case), the watchdog transmits the WDG\_RSTN reset signal to reset the system. (When the chip uses the built-in POR, the reset signal generated by the watchdog resets the chip and is output through the SYS\_RSTN\_OUT pin. When the chip uses the external POR, the reset signal generated by the watchdog is output to the external reset circuit through the SYS\_RSTN\_OUT pin, and then the reset signal generated by the reset circuit is input to the pin to reset the entire chip.) In this way, the system running status is monitored.

[Figure 3-6](#) shows the application block diagram of the watchdog.

**Figure 3-6** Application block diagram of the watchdog

## Function Principle

The watchdog works based on a 32-bit down counter. The initial value is loaded by [WDG\\_LOAD](#). When the watchdog clock is enabled, the count value is decremented by 1 on the rising edge of each count clock. When the count value reaches 0, the watchdog generates an interrupt. On the next rising edge of the count clock, the counter reloads the initial value from [WDG\\_LOAD](#) and continues to count in decremental mode.

If the count value of the counter reaches 0 for the second time but the CPU does not clear the watchdog interrupt, the watchdog transmits the reset signal [WDG\\_RSTN](#) and the counter stops counting.

You can enable or disable the watchdog by configuring [WDG\\_CONTROL](#) as required. That is, you can control the watchdog whether to generate interrupts and reset signals.

- When the interrupt generation function is disabled, the watchdog counter stops counting.
- When the interrupt generation function is enabled again, the watchdog counter counts starting from the preset value of [WDG\\_LOAD](#) instead of the last count value. Before an interrupt is generated, the initial value can be reloaded.

By configuring [WDG\\_LOCK](#), you can disable the operation of writing to the internal registers of the watchdog.

- Writing 0x1ACC\_E551 to [WDG\\_LOCK](#) to enable the write permission for all the registers of the watchdog.
- Writing any other value to [WDG\\_LOCK](#) to disable the write permission for all the registers of the watchdog except [WDG\\_LOCK](#).

This feature avoids modifications to the watchdog registers by software. Therefore, the watchdog operation is not terminated by mistake by software when an exception occurs.



In debugging mode, the watchdog is disabled automatically to avoid the intervention to the normal debugging.

### 3.8.4 Operating Mode

#### Configuring the Frequency of the Count Clock

The watchdog count clock is a 3 MHz clock.

The count time  $T_{WDG}$  of the watchdog is calculated as follows:  $T_{WDG} = \text{Value}_{WDG\_LOAD} \times (1/f_{clk})$

##### NOTE

The definition of each parameter in the preceding formula is as follows:

- $T_{WDG}$  indicates the count time of the watchdog.
- $\text{Value}_{WDG\_LOAD}$  indicates the initial count value of the watchdog.
- $f_{clk}$  indicates the frequency of the watchdog count clock.

The count time of the watchdog is about 0s to 1400s.

#### Initializing the System

The watchdog counter stops counting after the system power-on reset. Before the system is initialized, the watchdog must be initialized and enabled. To initialize the watchdog, perform the following steps:

- Step 1** Write to [WDG\\_LOAD](#) to set the initial count value.
- Step 2** Write to [WDG\\_CONTROL](#) to enable the interrupt mask function and start the watchdog counter.
- Step 3** Write to [WDG\\_LOCK](#) to lock the watchdog to avoid the watchdog settings being modified by the software by mistake.

----End

#### Processing an Interrupt

After an interrupt is received from the watchdog, the interrupt must be cleared in time and the initial count value must be reloaded to the watchdog to restart counting. A watchdog interrupt is processed as follows:

- Step 1** Write 0x1ACC\_E551 to [WDG\\_LOCK](#) to unlock the watchdog.
- Step 2** Write to [WDG\\_INTCLR](#) to clear the watchdog interrupt and load the initial count value to the watchdog to restart counting.
- Step 3** Write any other value rather than 0x1ACC\_E551 to [WDG\\_LOCK](#) to lock the watchdog.

----End



## Disabling the Watchdog

You can control the status of the watchdog by writing 0 or 1 to [WDG\\_CONTROL\[inten\]](#).

- 0: disabled
- 1: enabled

## 3.8.5 Register Summary

Each watchdog module has one group of registers. The registers of the watchdog modules are the same except for the base addresses.

[Table 3-14](#) describes watchdog registers.

**Table 3-14** Summary of watchdog registers (base address: 0x1103\_0000)

Offset Address	Register	Description	Page Number
0x0000	WDG_LOAD	Initial count value register	<a href="#">3-121</a>
0x0004	WDG_VALUE	Current count value register	<a href="#">3-122</a>
0x0008	WDG_CONTROL	Control register	<a href="#">3-122</a>
0x000C	WDG_INTCLR	Interrupt clear register	<a href="#">3-122</a>
0x0010	WDG_RIS	Raw interrupt register	<a href="#">3-123</a>
0x0014	WDG_MIS	Masked interrupt status register	<a href="#">3-123</a>
0x0C00	WDG_LOCK	Lock register	<a href="#">3-124</a>

## 3.8.6 Register Description

### WDG\_LOAD

WDG\_LOAD is an initial count value register. It is used to configure the initial count value of the internal counter of the watchdog. The initial count value must be greater than 0.

Offset Address: 0x0000 Total Reset Value: 0xFFFF\_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RW	wdg_load	Initial count value of the watchdog counter	0xFFFFFFFF



## WDG\_VALUE

WDG\_VALUE is a current count value register. It is used to read the current count value of the internal counter of the watchdog.

Offset Address: 0x0004 Total Reset Value: 0xFFFF\_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RO	wdogvalue	Current count value of the watchdog counter	0xFFFFFFFF

## WDG\_CONTROL

WDG\_CONTROL is a control register. It is used to enable or disable the watchdog and control the interrupt and reset functions of the watchdog.

Offset Address: 0x0008 Total Reset Value: x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	resen	Output enable of the watchdog reset signal 0: disabled 1: enabled	0x0
[0]	RW	inten	Output enable of the watchdog interrupt signal 0: The counter stops counting, the current count value remains unchanged, and the watchdog is disabled. 1: The counter, interrupt and watchdog are enabled.	0x0

## WDG\_INTCLR

WDG\_INTCLR is an interrupt clear register. It is used to clear watchdog interrupts so the watchdog can reload an initial value for counting. This register is write-only. When a value is written to this register, the watchdog interrupts are cleared. No written value is recorded in this register and no default reset value is defined.

Offset Address: 0x000C Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	WO	wdg_intclr	Writing any value to this register clears the watchdog interrupts and enables the watchdog to reload an initial count value from <a href="#">WDG_LOAD</a> to restart counting.	0x0000_0000

## WDG\_RIS

WDG\_RIS is a raw interrupt status register. It shows the raw interrupt status of the watchdog.

Offset Address: 0x0010 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	wdogris	Status of the raw interrupts of the watchdog. When the count value reaches 0, this bit is set to 1. 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## WDG\_MIS

WDG\_MIS is a masked interrupt status register. It shows the masked interrupt status of the watchdog.

Offset Address: 0x0014 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	wdogmis	Status of the masked interrupts of the watchdog 0: No interrupt is generated or the interrupt is masked. 1: An interrupt is generated.	0x0



## WDG\_LOCK

WDG\_LOCK is a lock register. It is used to control the write and read permissions for the watchdog registers.

Offset Address: 0x0C00 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wdg_lock	<p>Writing 0x1ACC_E551 to this register enables the write permission for all the registers.</p> <p>Writing other values disables the write permission for all the registers.</p> <p>When this register is read, the lock status rather than the written value of this register is returned.</p> <p>0x0000_0000: The write permission is available (unlocked).</p> <p>0x0000_0001: The write permission is unavailable (locked).</p>	0x00000000

## 3.9 RTC

### 3.9.1 Overview

The RTC is used to display the time in realtime and periodically generate alarms.

### 3.9.2 Features

The RTC has the following features:

- Provides one 16-bit day counter, 5-bit hour counter, 6-bit minute counter, 6-bit second counter, and 7-bit 10 ms counter.
- Provides a 100 Hz count clock.
- Supports the configurable initial count value.
- Supports the configurable count comparison value.
- Generates the timeout interrupt.
- Supports soft reset.
- Supports the fixed frequency-division mode.
- Stores user data in a 64-bit user register.



### 3.9.3 Function Description

The RTC has a 40-bit up counter for counting days, hours, minutes, seconds, and 10 ms. The initial values are loaded from [RTC\\_LR\\_10MS](#), [RTC\\_LR\\_S](#), [RTC\\_LR\\_M](#), [RTC\\_LR\\_H](#), [RTC\\_LR\\_D\\_L](#), and [RTC\\_LR\\_D\\_H](#). This section assumes that the counter is divided into the day counter, hour counter, minute counter, second counter, and 10 ms counter. When the count value is equal to the values of [RTC\\_MR\\_10MS](#), [RTC\\_MR\\_S](#), [RTC\\_MR\\_M](#), [RTC\\_MR\\_H](#), [RTC\\_MR\\_D\\_L](#), and [RTC\\_MR\\_D\\_H](#), the RTC generates an interrupt. Then, the counter continues to count in incremental mode on the rising edge of the next count clock.

By configuring [RTC\\_IMSC](#), you can allow or forbid the RTC to generate interrupts. Note the following two cases:

- When the function of generating interrupts is disabled, the RTC counter continues to count in incremental mode and no interrupts are generated. [RTC\\_MSC\\_INT](#) shows the status of masked interrupts and [RTC\\_RAW\\_INT](#) shows the status of raw interrupts.
- When the function of generating interrupts is enabled, the RTC counter still counts in incremental mode. When the count value is equal to the values of [RTC\\_MR\\_10MS](#), [RTC\\_MR\\_S](#), [RTC\\_MR\\_M](#), [RTC\\_MR\\_H](#), [RTC\\_MR\\_D\\_L](#), and [RTC\\_MR\\_D\\_H](#), the RTC generates an interrupt.

The RTC uses a 100 Hz count clock and a 16-bit day counter. The value of the day counter can be used to reckon the specific year, month, and day.

### 3.9.4 Operating Mode

#### 3.9.4.1 Count Clock Frequency

The RTC uses a 100 Hz count clock. The maximum RTC count time ( $T_{RTC}$ ) is calculated as follows:

$$T_{RTC} = 2^{16} = 65536 \text{ days}$$



$T_{RTC}$  is the RTC count time.

#### 3.9.4.2 Soft Reset

The RTC can be separately soft-reset by configuring [RTC\\_POR\\_N](#). After soft reset, the value of each RTC configuration register is restored to its default value. Therefore, these registers must be initialized again.

To soft-reset the RTC, perform the following steps:

**Step 1** Write 0 to [RTC\\_POR\\_N](#).

**Step 2** Wait 30 ms.

----End



### 3.9.4.3 Initializing the RTC

The system needs to initialize the RTC when the RTC is powered on for the first time. The initialization process is as follows:

- Step 1** Configure `RTC_POR_N` to reset the RTC.
  - Step 2** Wait 30 ms.
  - Step 3** Configure `RTC_IMSC` to set the interrupt mask bit of the RTC.
  - Step 4** Configure `RTC_MR_10MS`, `RTC_MR_S`, `RTC_MR_M`, `RTC_MR_H`, `RTC_MR_D_L`, and `RTC_MR_D_H` to set the RTC match value.
  - Step 5** Configure `RTC_LR_10MS`, `RTC_LR_S`, `RTC_LR_M`, `RTC_LR_H`, `RTC_LR_D_L`, and `RTC_LR_D_H` to set the initial count value of the RTC.
  - Step 6** Set `RTC_LOAD` to 1 to load the initial count value to the RTC counter.
  - Step 7** Wait 10 ms.
  - Step 8** Based on the 100 Hz count clock, the RTC counts starting from the values of `RTC_LR_10MS`, `RTC_LR_S`, `RTC_LR_M`, `RTC_LR_H`, `RTC_LR_D_L`, and `RTC_LR_D_H`. When the count value is equal to the values of `RTC_MR_10MS`, `RTC_MR_S`, `RTC_MR_M`, `RTC_MR_H`, `RTC_MR_D_L`, and `RTC_MR_D_H`, the RTC determines whether to generate an interrupt based on the settings of `RTC_IMSC`.
- End

### 3.9.4.4 Configuring the RTC Time

The procedure for configuring the RTC time is as follows:

- Step 1** Configure `RTC_LR_10MS`, `RTC_LR_S`, `RTC_LR_M`, `RTC_LR_H`, `RTC_LR_D_L`, and `RTC_LR_D_H` to set the initial RTC count value.
  - Step 2** Set `RTC_LOAD` to 1 to load the initial count value to the RTC counter.
  - Step 3** Wait 10 ms.
- End

### 3.9.4.5 Reading the RTC Time

The procedure for reading the RTC time is as follows:

- Step 1** Set `RTC_LOAD` to 0x2.
- Step 2** Read back the value of `RTC_LOAD`. If the value is not 0x0, repeat Step 2. If the value is 0x0, go to Step 3.
- Step 3** Wait 1 ms.
- Step 4** Read the time values from the registers `RTC_10MS_COUNT`, `RTC_S_COUNT`, `RTC_M_COUNT`, `RTC_H_COUNT`, `RTC_D_COUNT_L`, and `RTC_D_COUNT_H`.



----End

### 3.9.4.6 Handling Interrupts

If the system receives an interrupt from the RTC, the configured time is reached. Then user-defined operations can be performed. The RTC counter, however, still counts in incremental mode. RTC interrupts are handled as follows:

**Step 1** To clear an RTC interrupt, set [RTC\\_INT\\_CLR](#) to 1.

**Step 2** To continue to configure time, write a new match value to [RTC\\_MR\\_10MS](#), [RTC\\_MR\\_S](#), [RTC\\_MR\\_M](#), [RTC\\_MR\\_H](#), [RTC\\_MR\\_D\\_L](#), and [RTC\\_MR\\_D\\_H](#).

----End

### 3.9.5 Register Summary

[Table 3-15](#) describes the internal RTC registers.

**Table 3-15** Summary of internal RTC registers (base address: 0x1111\_0000)

Offset Address	Register	Description	Page
0x0000	RTC_10MS_COUNT	Count value register for the RTC 10 ms counter	<a href="#">3-129</a>
0x0004	RTC_S_COUNT	Count value register for the RTC second counter	<a href="#">3-130</a>
0x0008	RTC_M_COUNT	Count value register for the RTC minute counter	<a href="#">3-130</a>
0x000C	RTC_H_COUNT	Count value register for the RTC hour counter	<a href="#">3-130</a>
0x0010	RTC_D_COUNT_L	Lower-8-bit count value register for the RTC day counter	<a href="#">3-131</a>
0x0014	RTC_D_COUNT_H	Upper-8-bit count value register for the RTC day counter	<a href="#">3-131</a>
0x0018	RTC_MR_10MS	Match value register for the RTC 10 ms counter	<a href="#">3-131</a>
0x001C	RTC_MR_S	Match value register for the RTC second counter	<a href="#">3-132</a>
0x0020	RTC_MR_M	Match value register for the RTC minute counter	<a href="#">3-132</a>



Offset Address	Register	Description	Page
0x0024	RTC_MR_H	Match value register for the RTC hour counter	<a href="#">3-132</a>
0x0028	RTC_MR_D_L	Lower-8-bit match value register for the RTC day counter	<a href="#">3-132</a>
0x002C	RTC_MR_D_H	Upper-8-bit match value register for the RTC day counter	<a href="#">3-133</a>
0x0030	RTC_LR_10MS	Match value register for the RTC 10 ms counter	<a href="#">3-133</a>
0x0034	RTC_LR_S	Configured value register for the RTC second counter	<a href="#">3-133</a>
0x0038	RTC_LR_M	Configured value register for the RTC minute counter	<a href="#">3-134</a>
0x003C	RTC_LR_H	Configured value register for the RTC hour counter	<a href="#">3-134</a>
0x0040	RTC_LR_D_L	Lower-8-bit configured value register for the RTC day counter	<a href="#">3-134</a>
0x0044	RTC_LR_D_H	Upper-8-bit configured value register for the RTC day counter	<a href="#">3-135</a>
0x0048	RTC_LOAD	RTC configured value loading enable register	<a href="#">3-135</a>
0x004C	RTC_IMSC	RTC interrupt enable register	<a href="#">3-136</a>
0x0050	RTC_INT_CLR	RTC interrupt clear register	<a href="#">3-136</a>
0x0054	RTC_MSC_INT	RTC mask interrupt status register.	<a href="#">3-137</a>
0x0058	RTC_RAW_INT	RTC raw interrupt status register	<a href="#">3-137</a>
0x005C	RTC_CLK	RTC output clock select register	<a href="#">3-137</a>
0x0060	RTC_POR_N	RTC reset control register	<a href="#">3-138</a>
0x0068	RTC_UV_CTRL	RTC internal low-voltage detection control register	<a href="#">3-138</a>
0x006C	RTC_ANA_CTRL	RTC oscillating current control register	<a href="#">3-139</a>
0x0144	SDM_COEF_OUSID_E_H	External frequency division coefficient upper four bits register	<a href="#">3-139</a>
0x0148	SDM_COEF_OUSID_E_L	External frequency division coefficient lower eight bits register	<a href="#">3-140</a>



Offset Address	Register	Description	Page
0x014C	USER_REGISTER1	64-bit user register 1	<a href="#">3-140</a>
0x0150	USER_REGISTER2	64-bit user register 2	<a href="#">3-140</a>
0x0154	USER_REGISTER3	64-bit user register 3	<a href="#">3-141</a>
0x0158	USER_REGISTER4	64-bit user register 4	<a href="#">3-141</a>
0x015C	USER_REGISTER5	64-bit user register 5	<a href="#">3-141</a>
0x0160	USER_REGISTER6	64-bit user register 6	<a href="#">3-142</a>
0x0164	USER_REGISTER7	64-bit user register 7	<a href="#">3-142</a>
0x0168	USER_REGISTER8	64-bit user register 8	<a href="#">3-142</a>
0x0180	USER_DEFINE_REGISTER1	User-defined register 1	<a href="#">3-142</a>
0x0184	USER_DEFINE_REGI STER2	User-defined register 2	<a href="#">3-143</a>
0x0188	USER_DEFINE_REGI STER3	User-defined register 3	<a href="#">3-143</a>
0x018C	USER_DEFINE_REGI STER4	User-defined register 4	<a href="#">3-143</a>
0x0190	RTC_REG_LOCK1	RTC lock register 1	<a href="#">3-144</a>
0x0194	RTC_REG_LOCK2	RTC lock register 2	<a href="#">3-144</a>
0x0198	RTC_REG_LOCK3	RTC lock register 3	<a href="#">3-144</a>
0x019C	RTC_REG_LOCK4	RTC lock register 4	<a href="#">3-144</a>

### 3.9.6 RTC Internal Register Description

#### RTC\_10MS\_COUNT

RTC\_10MS\_COUNT is a count value register for the RTC 10 ms counter.

Offset Address: 0x0000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	Reserved	0x00000000
[6:0]	RO	rtc_10ms_count	RTC 10 ms counter value. It indicates the	0x00



Bits	Access	Name	Description	Reset
			currently counted time. Its unit is 10 ms. The value range is [0, 99].	

## RTC\_S\_COUNT

RTC\_S\_COUNT is a count value register for the RTC second counter.

Offset Address: 0x0004 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:0]	RO	rtc_s_count	RTC second counter value. It indicates the currently counted seconds. The value range is [0, 59].	0x00

## RTC\_M\_COUNT

RTC\_M\_COUNT is a count value register for the RTC minute counter.

Offset Address: 0x0008 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:0]	RO	rtc_m_count	RTC minute counter value. It indicates the currently counted minutes. The value range is [0, 59].	0x00

## RTC\_H\_COUNT

RTC\_H\_COUNT is a count value register for the RTC hour counter.

Offset Address: 0x000C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4:0]	RO	rtc_h_count	RTC hour counter value. It indicates the currently counted hours.	0x00



		The value range is [0, 23].	
--	--	-----------------------------	--

## RTC\_D\_COUNT\_L

RTC\_D\_COUNT\_L is a lower-8-bit count value register for the RTC day counter.

Offset Address: 0x0010      Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[7:0]	RO	rtc_d_count_l	Lower eight bits of the RTC day counter value. This field works with RTC_D_COUNT_H to indicate the currently counted days.  The value range is [0, 65535].	0x00

## RTC\_D\_COUNT\_H

RTC\_D\_COUNT\_H is an upper-8-bit count value register for the RTC day counter.

Offset Address: 0x0014      Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x00000000
[7:0]	RO	rtc_d_count_h	Upper eight bits of the RTC day counter value. This field works with RTC_D_COUNT_L to indicate the currently counted days.  The value range is [0, 65535].	0x00

## RTC\_MR\_10MS

RTC\_MR\_10MS is a match value register for the RTC 10 ms counter.

Offset Address: 0x0018      Total Reset Value: 0x0000\_007f

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	Reserved	0x00000000
[6:0]	RW	rtc_mr_10ms	Match value of the RTC 10 ms counter  The value range is [0, 99].	0x7F



## RTC\_MR\_S

RTC\_MR\_S is a match value register for the RTC second counter.

Offset Address: 0x001C      Total Reset Value: 0x0000\_003f

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:0]	RW	rtc_mr_s	Match value of the RTC second counter The value range is [0, 59].	0x3F

## RTC\_MR\_M

RTC\_MR\_M is a match value register for the RTC minute counter.

Offset Address: 0x0020      Total Reset Value: 0x0000\_003f

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:0]	RW	rtc_mr_m	Match value of the RTC minute counter The value range is [0, 59].	0x3F

## RTC\_MR\_H

RTC\_MR\_H is a match value register for the RTC hour counter.

Offset Address: 0x0024      Total Reset Value: 0x0000\_001f

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4:0]	RW	rtc_mr_h	Match value of the RTC hour counter The value range is [0, 23]	0x1f

## RTC\_MR\_D\_L

RTC\_MR\_D\_L is a lower-8-bit match value register for the RTC day counter.

Offset Address: 0x0028      Total Reset Value: 0x0000\_00ff



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	rtc_mr_d_l	Lower 8 bits of the match value of the RTC day counter. This field works with <a href="#">RTC_MR_D_H</a> to indicate the matched days. The value range is [0, 65535].	0xFF

## RTC\_MR\_D\_H

RTC\_MR\_D\_H is an upper-8-bit match value register for the RTC day counter.

Offset Address: 0x002C Total Reset Value: 0x0000\_00ff

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	rtc_mr_d_h	Upper 8 bits of the match value of the RTC day counter. This field works with <a href="#">RTC_MR_D_L</a> to indicate the matched days. The value range is [0, 65535].	0xFF

## RTC\_LR\_10MS

RTC\_LR\_10MS is a configured value register for the RTC 10 ms counter.

Offset Address: 0x0030 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	Reserved	0x00000000
[6:0]	RW	rtc_lr_10ms	Configured value of the RTC 10 ms counter The value range is [0, 99].	0x00

## RTC\_LR\_S

RTC\_LR\_S is a configured value register for the RTC second counter.

Offset Address: 0x0034 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:0]	RW	rtc_lr_s	Configured value of the RTC second counter The value range is [0, 59].	0x00

## RTC\_LR\_M

RTC\_LR\_M is a configured value register for the RTC minute counter.

Offset Address: 0x0038 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:0]	RW	rtc_lr_m	Configured value of the RTC minute counter The value range is [0, 59].	0x00

## RTC\_LR\_H

RTC\_LR\_H is a configured value register for the RTC hour counter.

Offset Address: 0x003C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4:0]	RW	rtc_lr_h	Configured value of the RTC hour counter The value range is [0, 23].	0x00

## RTC\_LR\_D\_L

RTC\_LR\_D\_L is a lower-8-bit configured value register for the RTC day counter.

Offset Address: 0x0040 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x00000000



Bits	Access	Name	Description	Reset
[7:0]	RW	rtc_lr_d_l	Lower eight bits of the configured value of the RTC day counter. This field works with <a href="#">RTC_LR_D_H</a> to indicate the configured day. The value range is [0, 65535].	0x00

## RTC\_LR\_D\_H

RTC\_LR\_D\_H is an upper-8-bit configured value register for the RTC day counter.

Offset Address: 0x0044 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:0]	RW	rtc_lr_d_h	Upper eight bits of the configured value of the RTC day counter. This field works with <a href="#">RTC_LR_D_L</a> to indicate the configured day. The value range is [0, 65535].	0x00

## RTC\_LOAD

RTC\_LOAD is an RTC configured value loading enable register.

Offset Address: 0x0048 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2]	RW	rtc_lock_bypass	RTC lock enable 0: enabled. The RTC count value (0x00–0x05) is updated only after the RTC is successfully locked. 1: disabled. The RTC count value (0x00–0x05) is updated in realtime.	0x0
[1]	RW	rtc_lock	RTC lock. If software writes 1, hardware is automatically cleared after the RTC is successfully locked. Note: This field is valid only when	0x0



			rtc_lock_bypass is 0.	
[0]	RW	rtc_load	RTC configured value loading enable. When the field is enabled, the RTC configured value will be loaded to the RTC accumulator. If software writes 1 to load the configured value, hardware will automatically set this field to 0 after successful loading.	0x0

## RTC\_IMSC

RTC\_IMSC is an RTC interrupt enable register.

Offset Address: 0x004C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2]	RW	rtc_imsc	RTC global interrupt output enable 0: disabled 1: enabled	0x0
[1]	RW	rtc_imsc_uv	Battery low-voltage detection interrupt output enable 0: disabled 1: enabled	0x0
[0]	RW	rtc_imsc_time	RTC timing interrupt output enable 0: disabled 1: enabled	0x0

## RTC\_INT\_CLR

RTC\_INT\_CLR is an RTC interrupt clear register.

Offset Address: 0x0050 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	WO	rtc_int_clr	RTC interrupt clear register. Writing any value by software clears the interrupt. Reading back the value has no effect.	0x0



## RTC\_MSC\_INT

RTC\_MSC\_INT is an RTC mask interrupt status register.

Offset Address: 0x0054 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RO	mask_int_uv	Status of the masked battery low-voltage detection interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	mask_int_time	Status of the masked RTC timing interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## RTC\_RAW\_INT

RTC\_RAW\_INT is an RTC raw interrupt status register.

Offset Address: 0x0058 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RO	raw_int_uv	Status of the raw battery low-voltage detection interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	raw_int_time	Status of the raw RTC timing interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## RTC\_CLK

RTC\_CLK is an RTC output clock select register.

Offset Address: 0x005C Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1:0]	RW	clk_out_sel	Output test clock of the RTC 00: crystal oscillator clock 01: corrected 100 Hz clock 1x: 1 Hz clock	0x0

## RTC\_POR\_N

RTC\_POR\_N is an RTC reset control register.

Offset Address: 0x0060 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	rtc_por_n	RTC reset. This field is automatically set to 1 after the RTC is successfully reset. 0: reset 1: not reset and working properly	0x1

## RTC\_UV\_CTRL

RTC\_UV\_CTRL is an RTC internal low-voltage detection control register.

Offset Address: 0x0068 Total Reset Value: 0x0000\_0004

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5]	RW	bat_uv_ctrl_en	Low-voltage detection enable 0: disabled 1: enabled	0x0
[4]	RW	bat_uv_ctrl_sel	Low-voltage detection source select 0: no filtering 1: filtering	0x0
[3:2]	RW	bat_uv_sel	Battery undervoltage detection threshold (recommended default value: <b>01</b> ) 00: 1.6 V	0x1



Bits	Access	Name	Description	Reset
			01: 1.8 V 10: 2 V 11: 2.2 V	
[1:0]	RW	sample_time	Low-voltage detection cycle 00: 1s 01: 1 minute 10: 10 minutes 11: 30 minutes	0x0

## RTC\_ANA\_CTRL

RTC\_ANA\_CTRL is the RTC oscillating current control register.

Locked by the RTC lock register, this register can be modified only when {RTC\_LOCK3, RTC\_LOCK2, RTC\_LOCK1, RTC\_LOCK0}=0x5A5AABCD.

Offset Address: 0x006C Total Reset Value: 0x05

Bits	Access	Name	Description	Reset
[7:0]	RW	rtc_ana_ctrl	RTC oscillating current control  0x00: Level 0 0x01: Level 1 0x02: Level 2 0x03: Level 3  Other values: reserved  Note: A larger level indicates a larger oscillating power supply.	0x5

## SDM\_COEF\_OUSIDE\_H

SDM\_COEF\_OUSIDE\_H is an external frequency division coefficient upper four bits register.

Offset Address: 0x0144 Total Reset Value: 0x0000\_0008

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW	sdm_coef_ouside_	Upper four bits of the frequency division	0x8



		h	coefficient in fixed frequency-division mode	
--	--	---	--	--

## SDM\_COEF\_OUSIDE\_L

SDM\_COEF\_OUSIDE\_L is an external frequency division coefficient lower eight bits register.

Offset Address: 0x0148 Total Reset Value: 0x0000\_001B

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	sdm_coef_ouside_l	Lower eight bits of the frequency division coefficient in fixed frequency-division mode  Note: When clock divider registers (addresses of 0x51 and 0x52) are read or written, you need to read or write to the upper four bits and then the lower eight bits continuously. The clock divider is calculated as follows: $(f - 32700) \times 30.52$ . f indicates the oscillation frequency of the external crystal and ranges from 32700 to 32799.	0x1B

## USER\_REGISTER1

USER\_REGISTER1 is 64-bit user register 1.

Offset Address: 0x014C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_register1	64-bit user register 1 corresponding to bit[7:0]	0x00

## USER\_REGISTER2

USER\_REGISTER2 is 64-bit user register 2.

Offset Address: 0x0150 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_register2	64-bit user register 2 corresponding to bit[15:8]	0x00

## USER\_REGISTER3

USER\_REGISTER3 is 64-bit user register 3.

Offset Address: 0x0154 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_register3	64-bit user register 3 corresponding to bit[23:16]	0x00

## USER\_REGISTER4

USER\_REGISTER4 is 64-bit user register 4.

Offset Address: 0x0158 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_register4	64-bit user register 4 corresponding to bit[31:24]	0x00

## USER\_REGISTERS5

USER\_REGISTER5 is 64-bit user register 5.

Offset Address: 0x015C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_register5	64-bit user register 5 corresponding to bit[39:32]	0x00



## USER\_REGISTER6

USER\_REGISTER6 is 64-bit user register 6.

Offset Address: 0x0160 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:0]	RW	user_register6	64-bit user register 6 corresponding to bit[47:40]	0x00

## USER\_REGISTER7

USER\_REGISTER7 is 64-bit user register 7.

Offset Address: 0x0164 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:0]	RW	user_register7	64-bit user register 7 corresponding to bit[55:48]	0x00

## USER\_REGISTER8

USER\_REGISTER8 is 64-bit user register 8.

Offset Address: 0x0168 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:0]	RW	user_register8	64-bit user register 8 corresponding to bit[63:56]	0x00

## USER\_DEFINE\_REGISTER1

USER\_DEFINE\_REGISTER1 is user-defined register 1.

Offset Address: 0x0180 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_define_register1	User-defined register 1	0x00

## USER\_DEFINE\_REGISTER2

USER\_DEFINE\_REGISTER2 is user-defined register 2.

Offset Address: 0x0184 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_define_register2	User-defined register 2	0x00

## USER\_DEFINE\_REGISTER3

USER\_DEFINE\_REGISTER3 is user-defined register 3.

Offset Address: 0x0188 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_define_register3	User-defined register 3	0x00

## USER\_DEFINE\_REGISTER4

USER\_DEFINE\_REGISTER4 is user-defined register 4.

Offset Address: 0x018C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	user_define_register4	User-defined register 4	0x00



## RTC\_REG\_LOCK1

RTC\_REG\_LOCK1 is RTC lock register 1.

Offset Address: 0x0190 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	rtc_reg_lock1	RTC register protection lock register 1	0x00

## RTC\_REG\_LOCK2

RTC\_REG\_LOCK2 is RTC lock register 2.

Offset Address: 0x0194 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	rtc_reg_lock2	RTC register protection lock register 2	0x00

## RTC\_REG\_LOCK3

RTC\_REG\_LOCK3 is RTC lock register 3.

Offset Address: 0x0198 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	rtc_reg_lock3	RTC register protection lock register 3	0x00

## RTC\_REG\_LOCK4

RTC\_REG\_LOCK4 is RTC lock register 4.

Offset Address: 0x019C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	rtc_reg_lock4	RTC register protection lock register 4	0x00



## 3.10 Power Management and Low-Power Mode Control

### 3.10.1 Overview

- In low-power mode, the power consumption of the chip is reduced effectively. The Hi3516CV610 reduces its power consumption in the following low-power control modes:
  - Clock gating and clock frequency adjustment

The clock disabling function is used to disable unnecessary clocks to reduce the power consumption of the chip. In addition, the frequency of the system working clock can be adjusted. That is, when the function requirement is met, you can adjust the clock frequency to reduce the power consumption of the chip.
  - Module low-power control

When a module is idle, it can be disabled or switched to low-power mode to reduce the power consumption.
  - DDR low-power control

The DDR controller dynamically manages the power consumption. You can enable this function to reduce the power consumption of the chip.
  - Selective voltage binning (SVB)
- Detection of the chip internal temperature

The chip integrates a temperature sensor (T-Sensor) to obtain the internal temperature.

### 3.10.2 Clock Gating and Clock Frequency Adjustment

#### Disabling Unnecessary Clocks

The system supports clock gating of each module. When a module is idle, its clock can be disabled to reduce chip power consumption. For details about the process, see the description in section "clock gating" of each module.

#### Adjusting the Working Frequency of the Module

The working frequencies of some modules can also be adjusted separately. This reduces the power consumption of the system further. For details about the clock source of each module, see section "[3.2.3 Clock Distribution](#)".



### 3.10.3 Module Low-Power Control

Each chip module supports low-power operating modes. The clocks of a module can be disabled by configuring the corresponding CRG control registers. The unused PLLs can be disabled by configuring corresponding registers to reduce the power consumption. For details, see section "[3.2 Clock](#)".

### 3.10.4 DDR Low-Power Control

For details about the low-power control of the DDRC, see "Low-Power Management" in chapter "DDRC."

### 3.10.5 SVB Function Description

The SVB technology further reduces power consumption by statically adjusting the voltage in realtime based on the chip technology, temperature, and circuit timings.

The chip contains the programmable PWM0 module. The peripheral regulator circuit can adjust the power voltage of the chip by using the SVB PWM pin output of the chip.

For details about the internal temperature detection, see section [3.10.6 "Tsensor\\_CTRL Function Description"](#).

### 3.10.6 Tsensor\_CTRL Function Description

The chip has an integrated T-Sensor and the detected temperature ranges from  $-40^{\circ}\text{C}$  ( $-40^{\circ}\text{F}$ ) to  $+125^{\circ}\text{C}$  ( $257^{\circ}\text{F}$ ). To enable the temperature sensor to collect data, perform the following steps:

**Step 1** Write 0x1 to PERI\_CRG4560[tsensor\_ctrl\_pclk\_cken] to enable the APB clock gating of the TSENSOR\_CTRL, and write **0x1** to PERI\_CRG4560[tsensor\_ctrl\_cken] to enable the clock gating of the TSENSOR\_CTRL.

**Step 2** Set the T-Sensor capture mode by configuring [TSENSOR\\_CTRL0](#)bit[30].

**Step 3** If the cyclic capture mode is used, set the capture period by configuring [TSENSOR\\_CTRL0](#)bit[27:20]. If the single capture mode is used, skip this step.

The formula for calculating the cyclic capture period is as follows:

$$T = N \times 2 \text{ ( ms)}$$

**N** is the value of [TSENSOR\\_CTRL0](#)bit[27:20].

**Step 4** Enable the T-Sensor by configuring [TSENSOR\\_CTRL0](#)bit[31] and start to collect the temperature.

**Step 5** Software reads the temperature code (hexadecimal value) collected by the T-Sensor.

In single capture mode, only temperature code 0 recorded in [TSENSOR\\_CTRL2](#) bit[9:0] is valid.



In cyclic capture mode, the latest eight temperature codes (codes 0–7) are recorded in [TSENSOR\\_CTRL2](#) bit[31:0] and [TSENSOR\\_CTRL5](#) bit[31:0]. The latest recorded temperature data is temperature code 0.

- Step 6** Calculate the corresponding temperature value based on the recorded temperature code.

The temperature value T (in °C) is calculated as follows:

Temperature = [(tsensor\_result – 127)/784]\*165 – 40 (°C). The value range of [tsensor\\_result](#) is [127, 911].

Note that tsensor\_result is the decimal value of the temperature code obtained in Step 4.

----End

### 3.10.7 Tsensor\_CTRL Register Summary

[Table 3-16](#) describes the Tsensor\_CTRL registers.

**Table 3-16** Summary of Tsensor\_CTRL registers (base address: 0x1102\_A000)

Offset Address	Register	Description	Page
0x0	TSENSOR_CTRL0	T-Sensor control register	<a href="#">3-148</a>
0x4	TSENSOR_CTRL1	T-Sensor status register	<a href="#">3-148</a>
0x8	TSENSOR_CTRL2	T-Sensor temperature record register 0	<a href="#">3-149</a>
0xC	TSENSOR_CTRL3	T-Sensor temperature record register 1	<a href="#">3-149</a>
0x10	TSENSOR_CTRL4	T-Sensor temperature record register 2	<a href="#">3-150</a>
0x14	TSENSOR_CTRL5	T-Sensor temperature record register 3	<a href="#">3-150</a>
0x20	TSENSOR_INT_MASK	T-Sensor interrupt mask register	<a href="#">3-150</a>
0x24	TSENSOR_INT_CLR	T-Sensor interrupt clear register	<a href="#">3-151</a>
0x28	TSENSOR_INT_RAW	T-Sensor raw interrupt status register	<a href="#">3-151</a>
0x2C	TSENSOR_INT_STAT	T-Sensor masked interrupt status register	<a href="#">3-152</a>



Offset Address	Register	Description	Page
0x40	TSENSOR_CTRL6	T-Sensor temperature record register 4	<a href="#">3-152</a>
0x44	TSENSOR_CTRL7	T-Sensor temperature record register 5	<a href="#">3-153</a>
0x48	TSENSOR_CTRL8	T-Sensor temperature record register 6	<a href="#">3-153</a>
0x4C	TSENSOR_CTRL9	T-Sensor temperature record register 7	<a href="#">3-153</a>

### 3.10.8 Tsensor\_CTRL Register Description

#### TSENSOR\_CTRL0

TSENSOR\_CTRL0 is the T-Sensor control register.

Offset Address: 0x0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31]	RW	tsensor_en	T-Sensor enable 0: disabled 1: enabled	0x0
[30]	RW	tsensor_monitor_en	T-Sensor cyclic temperature monitoring enable 0: one-time measurement 1: cyclic measurement	0x0
[29:28]	-	reserved	Reserved	0x0
[27:20]	RW	tsensor_monitor_period	T-Sensor cyclic temperature monitoring period. The timing reference is 2 ms.	0x00
[19:10]	RW	tsensor_uplimit	Temperature upper limit	0x000
[9:0]	RW	tsensor_lowlimit	Temperature lower limit	0x000

#### TSENSOR\_CTRL1

TSENSOR\_CTRL1 is the T-Sensor status register.



Offset Address: 0x4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:2]	RW	tsensor_temp_ct_sel	Time for the T-Sensor to convert the temperature code value  00: 0.512 ms 01: 0.256 ms 10: 1.024 ms 11: 2.048 ms  Note: The cycle time must meet the following requirement: $T_{\text{temperature cycle time}} > 16 \times T_{\text{code conversion time}}$ .	0x0
[1]	RO	tsensor_low_warning	Temperature underflow warning, active high	0x0
[0]	RO	tsensor_up_warning	Temperature overflow warning, active high	0x0

## TSENSOR\_CTRL2

TSENSOR\_CTRL2 is T-Sensor temperature record register 0.

Offset Address: 0x8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RO	tsensor_result1	Temperature record 1	0x000
[15:10]	-	reserved	Reserved	0x00
[9:0]	RO	tsensor_result0	Temperature record 0	0x000

## TSENSOR\_CTRL3

TSENSOR\_CTRL3 is T-Sensor temperature record register 1.

Offset Address: 0xC Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00



Bits	Access	Name	Description	Reset
[25:16]	RO	tsensor_result3	Temperature record 3	0x000
[15:10]	-	reserved	Reserved	0x00
[9:0]	RO	tsensor_result2	Temperature record 2	0x000

## TSENSOR\_CTRL4

TSENSOR\_CTRL4 is T-Sensor temperature record register 2.

Offset Address: 0x10 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RO	tsensor_result5	Temperature record 5	0x000
[15:10]	-	reserved	Reserved	0x00
[9:0]	RO	tsensor_result4	Temperature record 4	0x000

## TSENSOR\_CTRL5

TSENSOR\_CTRL5 is T-Sensor temperature record register 3.

Offset Address: 0x14 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RO	tsensor_result7	Temperature record 7	0x000
[15:10]	-	reserved	Reserved	0x00
[9:0]	RO	tsensor_result6	Temperature record 6	0x000

## TSENSOR\_INT\_MASK

TSENSOR\_CTRL5 is the T-Sensor interrupt mask register.

Offset Address: 0x20 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	tsensor0_low_warning_int_mask	Raw interrupt mask control for the temperature underflow warning 0: interrupt masked 1: interrupt enabled	0x0
[0]	RW	tsensor0_up_warning_int_mask	Raw interrupt mask control for the temperature overflow warning 0: interrupt masked 1: interrupt enabled	0x0

## TSENSOR\_INT\_CLR

TSENSOR\_CTRL5 is the T-Sensor interrupt clear register.

Offset Address: 0x24 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	WO	tsensor0_low_warning_int_clr	Interrupt clear for the temperature underflow warning. Writing 1 clears this bit.	0x0
[0]	WO	tsensor0_up_warning_int_clr	Interrupt clear for the temperature overflow warning. Writing 1 clears this bit.	0x0

## TSENSOR\_INT\_RAW

TSENSOR\_INT\_RAW is the T-Sensor raw interrupt status register.

Offset Address: 0x28 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RO	tsensor0_low_warning_int_raw	Raw interrupt for the temperature underflow warning 0: No interrupt is generated. 1: An interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
[0]	RO	tsensor0_up_warning_int_raw	Raw interrupt for the temperature overflow warning 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## TSENSOR\_INT\_STAT

TSENSOR\_INT\_STAT is the T-Sensor masked interrupt status register.

Offset Address: 0x2C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RO	tsensor0_low_warning_int	Interrupt status of the masked temperature underflow warning 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	tsensor0_up_warning_int	Interrupt status of the masked temperature overflow warning 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## TSENSOR\_CTRL6

TSENSOR\_CTRL6 is the T-Sensor temperature record register 4.

Offset Address: 0x40 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RO	tsensor_result9	Temperature record 9	0x000
[15:10]	-	reserved	Reserved	0x00
[9:0]	RO	tsensor_result8	Temperature record 8	0x000



## TSENSOR\_CTRL7

TSENSOR\_CTRL7 is the T-Sensor temperature record register 5.

Offset Address: 0x44 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RO	tsensor_result11	Temperature record 11	0x000
[15:10]	-	reserved	Reserved	0x00
[9:0]	RO	tsensor_result10	Temperature record 10	0x000

## TSENSOR\_CTRL8

TSENSOR\_CTRL8 is the T-Sensor temperature record register 6.

Offset Address: 0x48 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RO	tsensor_result13	Temperature record 13	0x000
[15:10]	-	reserved	Reserved	0x00
[9:0]	RO	tsensor_result12	Temperature record 12	0x000

## TSENSOR\_CTRL9

TSENSOR\_CTRL9 is the T-Sensor temperature record register 7.

Offset Address: 0x4C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RO	tsensor_result15	Temperature record 15	0x000
[15:10]	-	reserved	Reserved	0x00
[9:0]	RO	tsensor_result14	Temperature record 14	0x000



# Contents

<b>4 Memory Interfaces .....</b>	<b>4-1</b>
4.1 DDRC.....	4-1
4.1.1 Overview.....	4-1
4.1.2 Features .....	4-1
4.1.3 Function Description .....	4-2
4.1.4 Operating Mode .....	4-7
4.2 Flash Memory Controller.....	4-8
4.2.1 Overview.....	4-8
4.2.2 Features .....	4-8
4.2.3 Function Description .....	4-10
4.2.4 Working Process .....	4-18
4.2.5 Data Structures (SPI NAND Flash) .....	4-23
4.2.6 ECC Mode Selection .....	4-26
4.2.7 FMC Registers.....	4-27
4.2.8 FMC Register Description.....	4-29



## Figures

<b>Figure 4-1</b> Interconnection between the DDRC and one 16-bit DDR2 SDRAM in single -channel mode.....	4-3
<b>Figure 4-2</b> Interconnection between the DDRC and one 16-bit DDR3/3L SDRAM in single-channel mode.....	4-4
<b>Figure 4-3</b> Write timing of the standard SPI .....	4-10
<b>Figure 4-4</b> Read timing of the standard SPI .....	4-10
<b>Figure 4-5</b> Timing of the dual-output/dual-input SPI.....	4-11
<b>Figure 4-6</b> STR timing of the dual I/O SPI.....	4-11
<b>Figure 4-7</b> DTR timing of the dual I/O SPI.....	4-12
<b>Figure 4-8</b> Timing of the quad-output/quad-input SPI .....	4-12
<b>Figure 4-9</b> Timing of the quad I/O SPI .....	4-13
<b>Figure 4-10</b> DTR timing of the quad I/O SPI .....	4-13
<b>Figure 4-11</b> SPI output timing .....	4-14
<b>Figure 4-12</b> Example of erase operation process.....	4-20
<b>Figure 4-13</b> Data structure when the page size is 2 KB in 4-bit ECC mode .....	4-25
<b>Figure 4-14</b> Data structure when the page size is 4 KB in 4-bit ECC mode .....	4-25
<b>Figure 4-15</b> Data structure when the page size is 2 KB in 8-bit ECC mode .....	4-26
<b>Figure 4-16</b> Data structure when the page size is 4 KB in 8-bit ECC mode .....	4-26
<b>Figure 4-17</b> Data structure when the page size is 2 KB in 24-bit ECC mode.....	4-26
<b>Figure 4-18</b> Data structure when the page size is 4 KB in 24-bit ECC mode.....	4-26



## Tables

<b>Table 4-1</b> Command truth values of the DDR2.....	4-5
<b>Table 4-2</b> Command truth values of the DDR3/3L SDRAM.....	4-6
<b>Table 4-3</b> Address allocation of the SPI NAND flash .....	4-14
<b>Table 4-4</b> Data structure length in various non-ECC0 modes.....	4-24
<b>Table 4-5</b> Summary of FMC registers (base address: 0x1000_0000) .....	4-27



# 4 Memory Interfaces

## 4.1 DDRC

### 4.1.1 Overview

The DDR synchronous dynamic random access memory (SDRAM) controller (DDRC) controls access to the dynamic memory of the DDR2/3/3L SDRAMs.

#### NOTE

- In this document, DDR2 SDRAM, DDR3 SDRAM, and DDR3L SDRAM are together abbreviated as DDR2/3/3L SDRAM.
- There are eight Hi3516CV610 models based on functions: 00B, 10B, 20B, 00S, 20S, 00G, and 20G.

### 4.1.2 Features

#### The DDRC has the following features

Hi3516CV610-10B:

- Maximum storage capacity of 512 Mb.
- Only little-endian mode.
- The maximum working frequency of the DDR2 SDRAM interface is 666.5 MHz, that is, the data rate is 1.333 Gbit/s.
- Various DDR2 SDRAM low-power modes.
- Single-channel.

Hi3516CV610-20B/20S/20G:

- Maximum storage capacity of 1 Gb.
- Only little-endian mode.
- The maximum working frequency of the DDR3/3L SDRAM interface is 1066.5 MHz, that is, the data rate is 2.133 Gbit/s.



- Various DDR3/3L SDRAM low-power modes.
- Single-channel.

Hi3516CV610-00B/00S/00G:

- Maximum storage capacity of 4 Gb.
- Only little-endian mode.
- External 16-bit DDR3/DDR3L SDRAM.
- The maximum working frequency of the DDR3/3L SDRAM interface is 1066.5 MHz, that is, the data rate is 2.133 Gbit/s.
- Various DDR3/3L SDRAM low-power modes
- Single-channel.

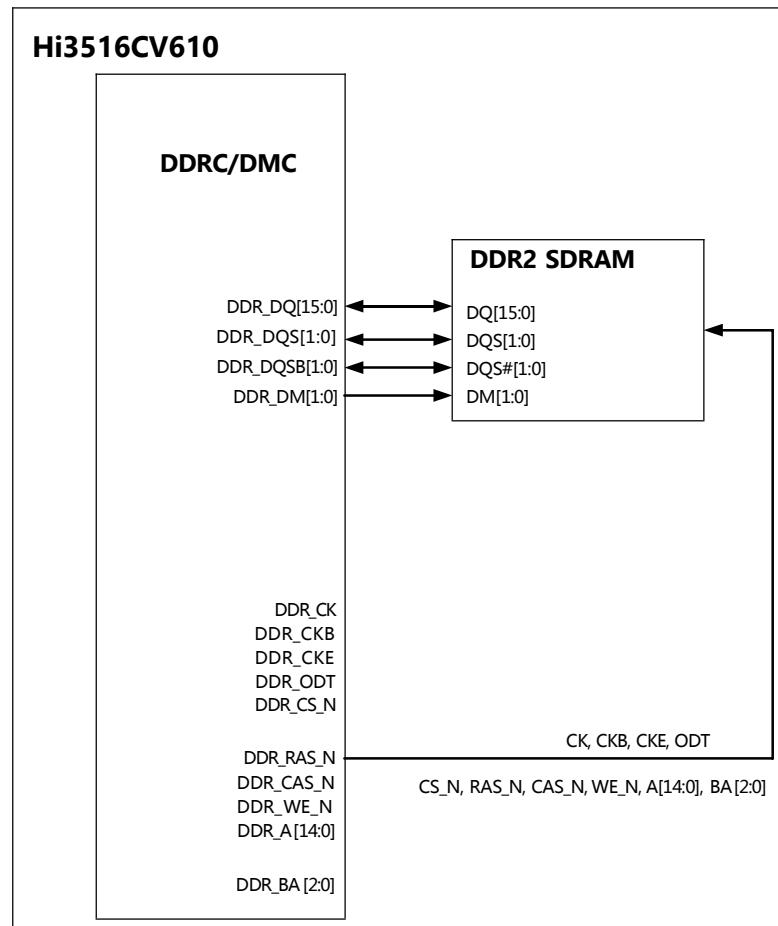
## 4.1.3 Function Description

### 4.1.3.1 Application Block Diagram

The DDRC enables the master devices of the SoC such as the CPU to access the DDR2/3/3L SDRAM. It supports the DDR2/3/3L SDRAM that complies with the JEDEC standard after the timing parameter registers of the DDRC are configured by using the CPU.

The DDRC is 16-bit per channel.[Figure 4-1](#) and [Figure 4-2](#) shows the interconnection diagrams.

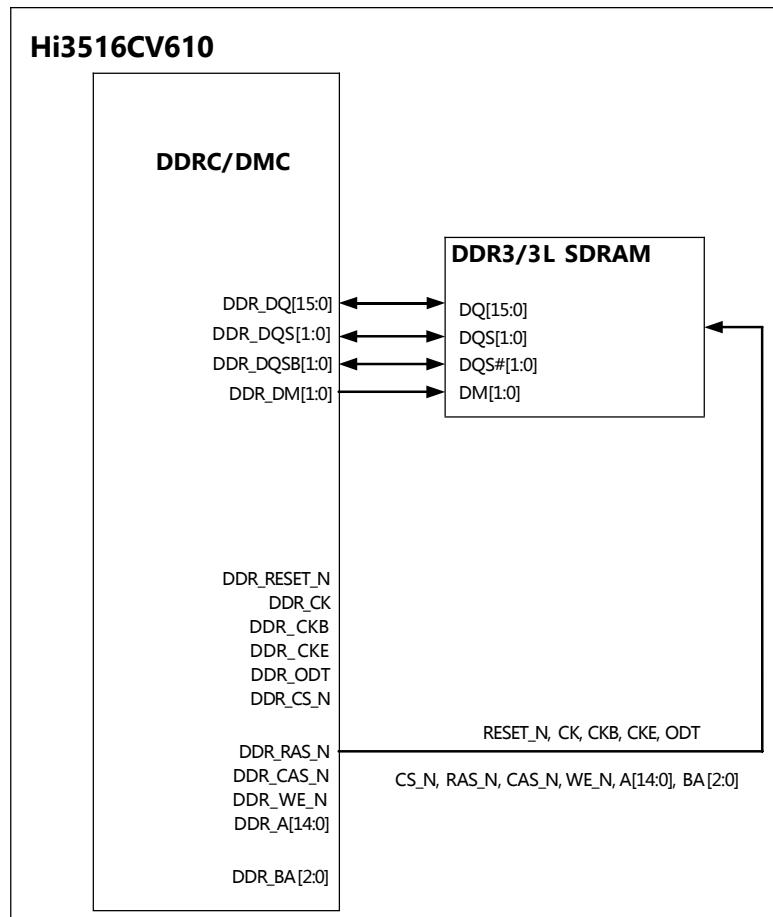
**Figure 4-1** Interconnection between the DDRC and one 16-bit DDR2 SDRAM in single-channel mode



### NOTE

In single-channel mode, the 16-bit DDR2 SDRAM consists of one piece of 16-bit memory. The DDRC/DMC command control signals (DDR\_CS\_N, DDR\_CK, DDR\_CKB, DDR\_CKE, DDR\_RAS\_N, DDR\_CAS\_N, DDR\_WE\_N, DDR\_A[14:0], and DDR\_BA[2:0]) connect to the command control signals of the DDR2 SDRAM, respectively. The DDRC command control bus uses the one-drive-one connection mode.

**Figure 4-2** Interconnection between the DDRC and one 16-bit DDR3/3L SDRAM in single-channel mode



#### NOTE

In single-channel mode, the 16-bit DDR3/3L SDRAM consists of one piece of 16-bit memory. The DDRC/DMC command control signals (DDR\_CS\_N, DDR\_CK, DDR\_CKB, DDR\_CKE, DDR\_RESET\_N, DDR\_RAS\_N, DDR\_CAS\_N, DDR\_WE\_N, DDR\_A[14:0], and DDR\_BA[2:0]) connect to the command control signals of the DDR3/3L SDRAM, respectively. The DDRC command control bus uses the one-drive-one connection mode.

#### 4.1.3.2 Function Implementation

As the timings of the DDRC interface comply with the JEDEC standard, the DDRC can access (read or write) data in the DDR2/3/3L SDRAM and control the status of the DDR2/3/3L SDRAM (including automatic refresh and low power control) by sending the command words of the DDR2/3/3L SDRAM.

#### Command Truth Value Table

The DDRC can read from and write to the DDR2 SDRAM as well as control command words.



For details about the truth table of DDR2 SDRAM, see the JEDEC DDR2 standard and flash data sheet.

[Table 4-1](#) lists the command truth values of the DDR2 SDRAM. For details, see the JEDEC standard and component data sheet.

**Table 4-1** Command truth values of the DDR2

FUNCTION	CKE	CSN	RASN	CASN	WEN	ADDR			BA
						12:11	AP(10)	9:0	
DESELECT	H	H	X	X	X	X	X	X	X
ACTIVE	H	L	H	H	H	Row Address (RA)			BA
READ	H	L		L	H	V	V	V	BA
WRITE	H	L	H	L	L	V	V	V	BA
PRECHARGE	H	L	L	H	L	V	L	V	BA
PRECHARGE ALL	H	L	L	H	L	V	H	V	V
AUTO REFRESH	H	L	L	L	H	V	V	V	V
SELF REFRESH ENTRY	H->L	L	L	L	H	V	V	V	V
SELF REFRESH EXIT	L->H	L	H	H	H	V	V	V	V
MODE REGISTER SET	H	L	L	L	L	V	V	V	V

H: high level; L: low level; V: valid; X: ignored.

The DDRC supports read, write, and control command words of the DDR3/3L SDRAM.

[Table 4-2](#) lists the command truth values of the DDR3/3L SDRAM. For details, see the JEDEC standard and component data sheet.



**Table 4-2** Command truth values of the DDR3/3L SDRAM

FUNCTION	CKE	CSN	RASN	CASN	WEN	ADDR			BA
						14:11	AP(10)	9:0	
DESELECT	H	H	X	X	X	X	X	X	X
ACTIVE	H	L	L	H	H	Row address			BA
READ	H	L	H	L	H	V	V	V	BA
WRITE	H	L	H	L	L	V	V	V	BA
PRECHARGE	H	L	L	H	L	V	L	V	BA
PRECHARGE ALL	H	L	L	H	L	V	H	V	V
AUTO REFRESH	H	L	L	L	H	V	V	V	V
SELF REFRESH ENTRY	H->L	L	L	L	H	V	V	V	V
SELF REFRESH EXIT	L->H	L	H	H	H	V	V	V	V
MODE REGISTER SET	H	L	L	L	L	V	V	V	V
ZQCL	H	L	H	H	L	V	H	V	V
ZQCS	H	L	H	H	L	V	L	V	V

H: high level; L: low level; V: valid; X: ignored.

## Auto Refresh

The DDRC refreshes the DDR2/3/3L SDRAM by automatically generating a periodical auto refresh command. At ambient temperature, the auto refresh period of the DDR2/3/3L SDRAM is 7.8  $\mu$ s. The actual value is provided in the component data sheet.

The SDRAM components may have different requirements on the auto-refresh cycle with the change of the temperature. For details, see the SDRAM user manual.



## Low-Power Management

The DDRC supports two modes of low-power management: common low-power mode and auto-refresh low-power mode.

When the SDRAM low-power mode is enabled, and the system is idle (the DDR SDRAM is not read or written through the DDRC bus interface for a period), the DDR2/3/3L SDRAM enters the common low-power mode automatically.

To switch the system mode to standby mode, you can force the DDR2/3/3L SDRAM to enter the auto-refresh mode. In this mode, the power consumption of the DDR2/3/3L SDRAM is minimized, data in the DDR2/3/3L SDRAM is retained, but the system cannot access the DDR2/3/3L SDRAM.

## Security Function

The DDRC provides the security function. When the security function is enabled, the bus error is returned for unauthorized read and write access to the bus, and the DDR SDRAM data is not overwritten or read.

The DDRC provides read and write protection for the secure function register. Only the secure CPU can read or write the register.

For details, see the *Hi35xxVxxx Security Subsystem User Guide*.

## Traffic Statistics and Command Latency Statistics

The DDRC supports traffic statistics. The interface read and write traffic statistics can be collected. The DDRC can collect the read and write traffic statistics of a specific ID, or the overall read and write traffic statistics. The DDRC can also collect DDR SDRAM interface usage statistics. The statistic counter supports the continuous counts and one-time count. When counting continuously, the counter is the non-saturating counter, and it is wrapped when the maximum count is reached to facilitate the continuous count. Therefore, the system needs to read the count value before the counter is wrapped. When counting for only one time, the counter stops counting when the statistic time elapses. The system can obtain the instantaneous traffic and latency by using this function.

### 4.1.4 Operating Mode

#### 4.1.4.1 Soft Reset

The DDRC does not support separate soft reset. It can be reset only by a global soft reset. After reset, the DDR2/3/3L SDRAM must be reinitialized according to the following initialization processes.

#### 4.1.4.2 Initializing the DDR SDRAM

After power-on, the system can access the DDR SDRAM only when the DDR2/3/3L SDRAM is initialized. Before initializing the DDR SDRAM, note the following:



- Power on the DDR2/3/3L SDRAM by following the JEDEC standard.
  - Initialize the DDR2/3/3L SDRAM after the system runs in normal mode.
- To initialize the whole DDR sub-system, perform the following steps:

- Step 1** Configure the DDR SDRAM address region mapping and address region attribute.
- Step 2** Configure the operating mode, clock frequency, and timing parameters of the DMC.
- Step 3** Configure the operating mode and time parameters of the DDR PHY, DDR PHY I/O drive, ODT impedance, and the delay parameter for the read and write command channels.
- Step 4** Initialize the DDR PHY.
- Step 5** Wait until the DDR PHY is initialized.
- Step 6** Configure the DMC to enable the auto-refresh command transfer function.

----End

After the preceding steps are complete, the DDR2/3/3L SDRAM works properly.

#### NOTE

The register values may vary according to the DDR SDRAM type, but the procedure is the same.

## 4.2 Flash Memory Controller

### 4.2.1 Overview

The flash memory controller (FMC) provides memory controller interfaces for connecting to external SPI NAND flash or SPI NOR flash memories to access data.

### 4.2.2 Features

The FMC has the following features:

- Supports a 4 KB + 256 bytes on-chip buffer for improving the read speed.
- Supports two external CS (for connecting to the SPI NAND flash or SPI NOR flash memory).
- Supports the SPI NOR flash and SPI NAND flash.
- Supports five types of SPIs, including the standard SPI, dual-output/dual-input SPI, quad-output/quad-input SPI, dual I/O SPI, and quad I/O SPI.
- Supports the SPI NOR flash with the following specifications:
  - Data reading in SPI NOR DTR/STR mode



- Maximum capacity of 32 MB
- Supports the SPI NAND flash with the following specifications:
  - Data reading in SPI NAND DTR/STR mode
  - 2 KB or 4 KB page size
  - 64-page or 128-page block
  - Maximum capacity of 512 MB
- Allows the system to boot from the SPI NOR flash memory, SPI NAND flash memory, and CS0.
  - Provides 1 MB boot space.
  - Automatically sends the reset command of the flash and then performs data read operations in boot mode. The reset command is sent only in SPI NAND flash mode.
  - Supports the fast read command. The frequency can be increased to 99 MHz at most for read operations.
  - Automatically skips bad blocks (for the SPI NAND flash). A maximum of four consecutive bad blocks can be skipped.
  - Supports the adaptive boot function (for the SPI NAND flash). The FMC automatically obtains the correct page size, error checking and correction (ECC) type, and block size.
  - Supports the 1-wire and 4-wire boot modes for the SPI NAND flash and 1-wire boot mode for the SPI NOR flash.
  - Supports boot from the SPI NOR flash in 3-byte or 4-byte address mode.
- Supports the direct memory access (DMA) read and write functions of the SPI NOR flash and SPI NAND flash.
  - Supports the read and write operations in internal DMA mode for the SPI NAND flash. You can write only the entire page and read the entire page or only the control information (read only the OOB).
  - Supports the DMA read and write operations for the SPI NOR flash. The length of the data read or written can be configured.
- Supports manual configuration of command combinations only in ECC0 mode using registers.
- Supports the ECC function (only for the SPI NAND flash).
- Supports seven types of interrupts: operation completion interrupt, programming failure interrupt, correctable ECC error interrupt, ECC alarm interrupt, uncorrectable ECC error interrupt, bus operation error interrupt, and DMA transfer error interrupt.
- Supports the low-power mode and automatic clock gating.

## 4.2.3 Function Description

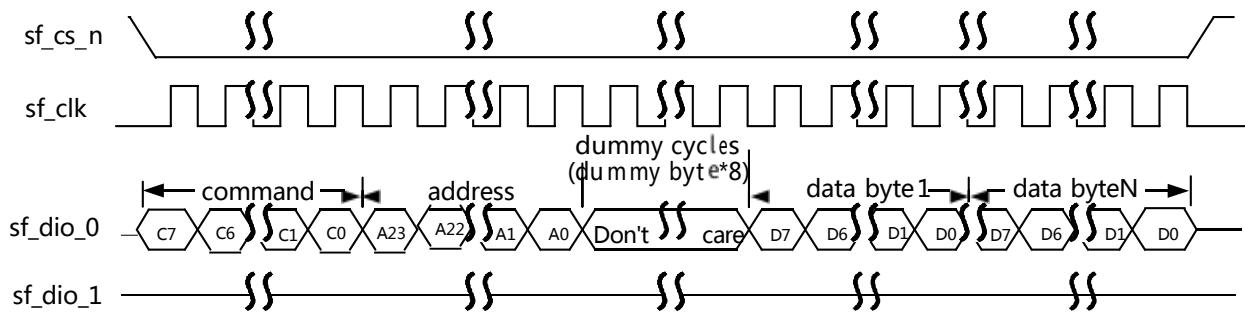
### 4.2.3.1 Interfaces

The FMC supports five types of SPIs, including the standard SPI, dual-output/dual-input SPI, quad-output/quad-input SPI, dual I/O SPI, and quad I/O SPI. The standard SPI, dual I/O SPI, and quad I/O SPI support the STR and DTR modes.

#### Standard SPI

The standard SPI is connected to a 1-bit data input line and a 1-bit data output line. [Figure 4-3](#) shows the write timing of the standard SPI.

**Figure 4-3** Write timing of the standard SPI

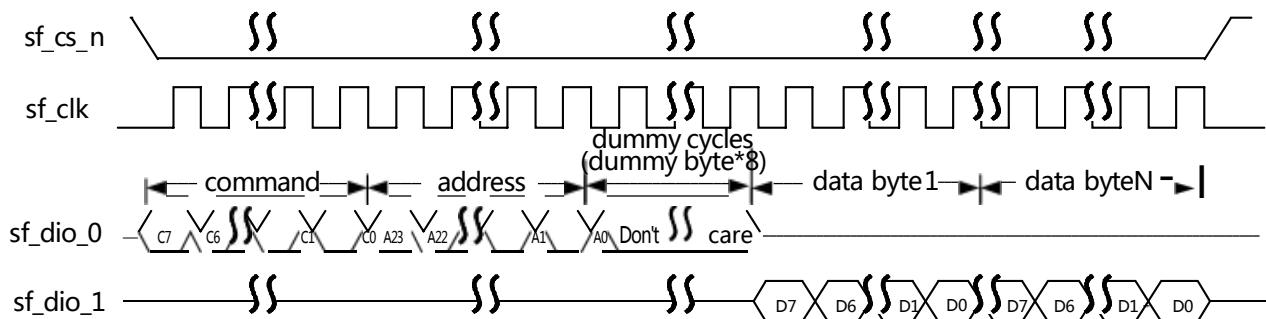


Note the following:

- The command cycles, address cycles, and dummy cycles are output in single-bit serial mode through the sf\_dio\_0 line.
- Data is output in single-bit serial mode through the sf\_dio\_0 line.

[Figure 4-4](#) shows the read timing of the standard SPI.

**Figure 4-4** Read timing of the standard SPI



Note the following:

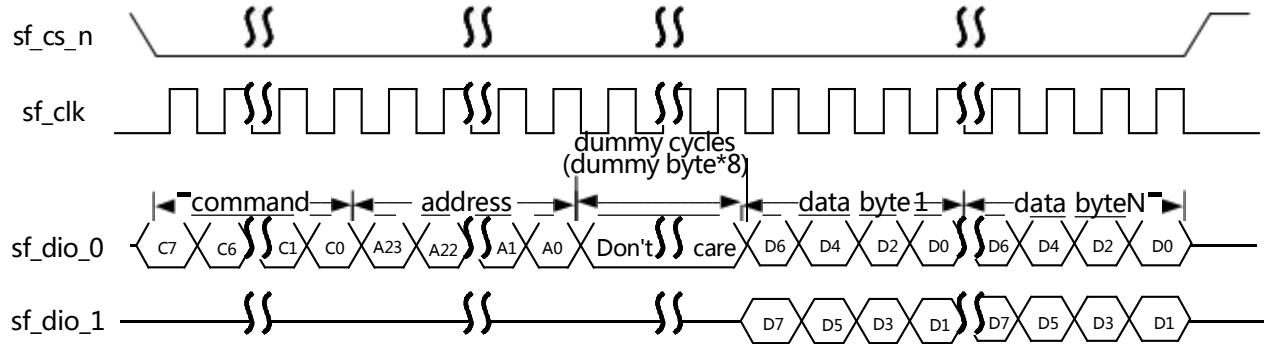


- The command cycles, address cycles, and dummy cycles are output in single-bit serial mode through the sf\_dio\_0 line.
- Data is input in single-bit serial mode through the sf\_dio\_1 line.

## Dual-Output/Dual-Input SPI

[Figure 4-5](#) shows the timing of the dual-output/dual-input SPI.

**Figure 4-5** Timing of the dual-output/dual-input SPI



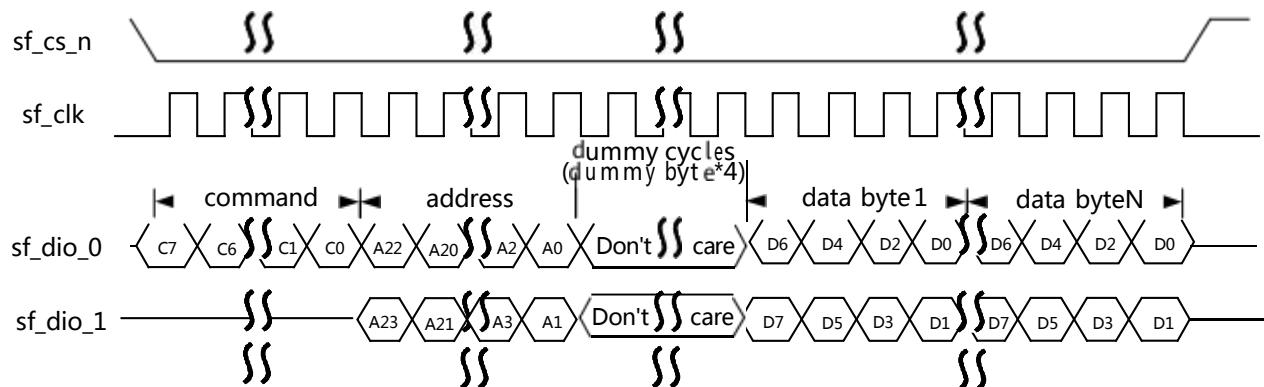
Note the following:

- The command cycles, address cycles, and dummy cycles are output in single-bit serial mode through the sf\_dio\_0 line.
- The data bytes are output (written) or input (read) in 2-bit mode through the sf\_dio\_0 or sf\_dio\_1 line.

## Dual I/O SPI

[Figure 4-6](#) and [Figure 4-7](#) show the timings of the dual I/O SPI.

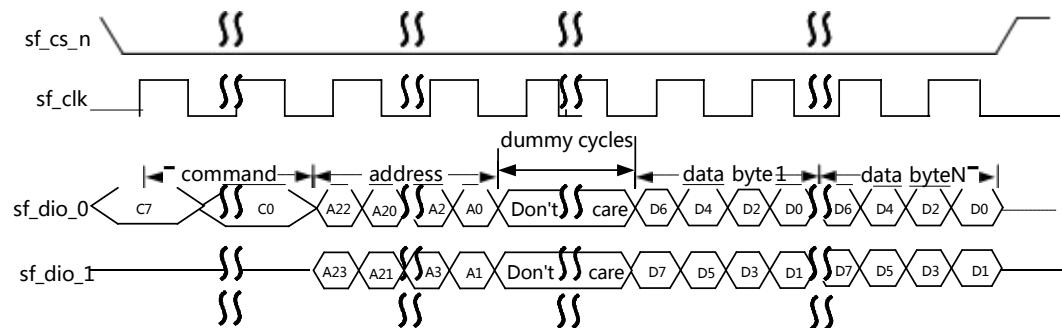
**Figure 4-6** STR timing of the dual I/O SPI



Note the following:

- The command cycles are output in single-bit serial mode through the sf\_dio\_0 line.
- The address cycles and dummy cycles are output in 2-bit mode through the sf\_dio\_0 or sf\_dio\_1 line.
- The data bytes are output (written) or input (read) in 2-bit mode through the sf\_dio\_0 or sf\_dio\_1 line.

**Figure 4-7** DTR timing of the dual I/O SPI



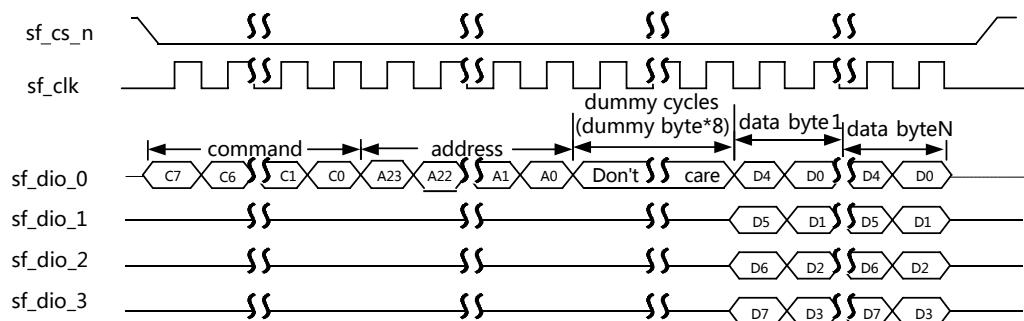
Note the following:

- The command cycles are output in single-bit serial mode through the sf\_dio\_0 line.
- The address cycles and dummy cycles are output in 2-bit dual-edge active mode through the sf\_dio\_0 or sf\_dio\_1 line.
- Data is input (read) in 2-bit dual-edge active mode through the sf\_dio\_0 or sf\_dio\_1 line.
- Only the read operation is supported in DTR mode.

## Quad-Output/Quad-Input SPI

**Figure 4-8** shows the timing of the quad-output/quad-input SPI.

**Figure 4-8** Timing of the quad-output/quad-input SPI



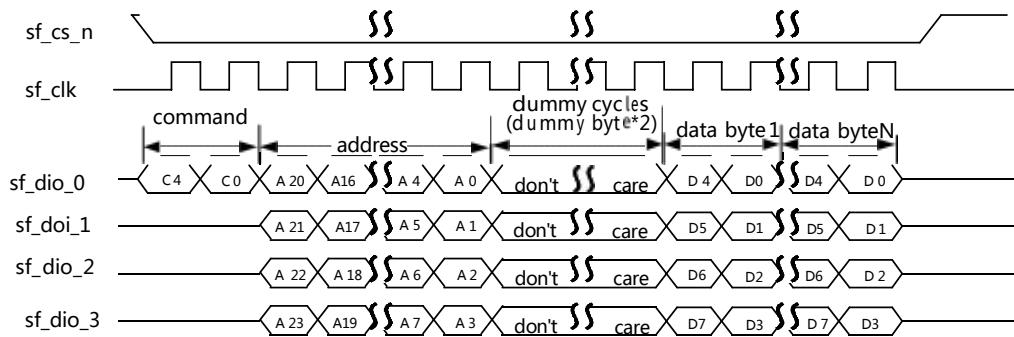
Note the following:

- The command cycles, address cycles, and dummy cycles are output in single-bit serial mode through the sf\_dio\_0 line.
- Data is output (written) or input (read) in 4-bit mode through the sf\_dio\_0, sf\_dio\_1, sf\_dio\_2, or sf\_dio\_3 line.

## Quad I/O SPI

[Figure 4-9](#) and [Figure 4-10](#) show the timings of the quad I/O SPI.

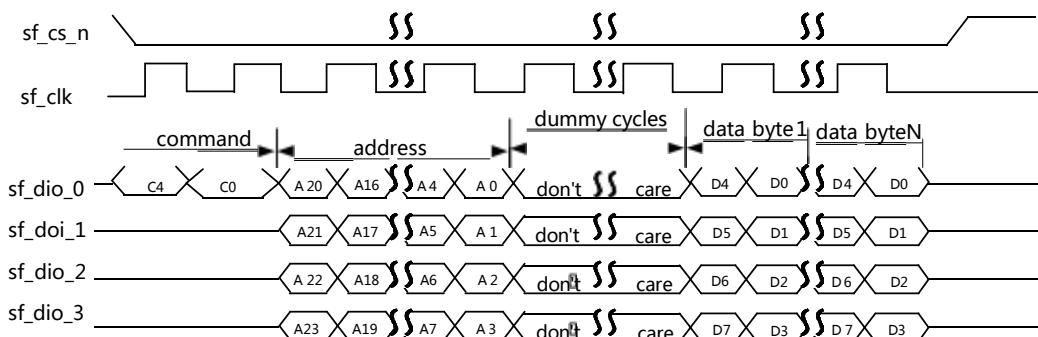
**Figure 4-9** Timing of the quad I/O SPI



Note the following:

- The command cycles are output in single-bit serial mode through the sf\_dio\_0 line.
- The address cycles and dummy cycles are output in 4-bit mode through the sf\_dio\_0, sf\_dio\_1, sf\_dio\_2, or sf\_dio\_3 line.
- The data bytes are output (written) or input (read) in 4-bit mode through the sf\_dio\_0, sf\_dio\_1, sf\_dio\_2, or sf\_dio\_3 line.

**Figure 4-10** DTR timing of the quad I/O SPI



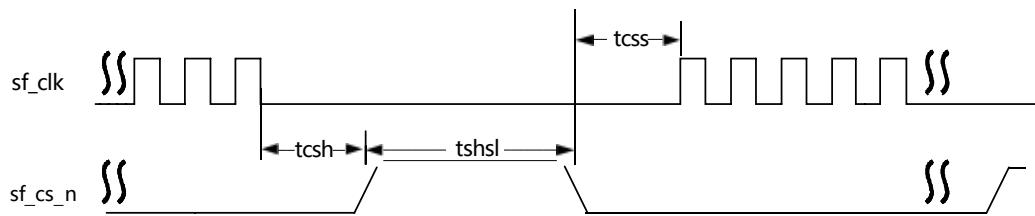
Note the following:

- The command cycles are output in single-bit single-edge data serial mode through the sf\_dio\_0 line.
- The address cycles and dummy cycles are output in 4-bit dual-edge mode through the sf\_dio\_0, sf\_dio\_1, sf\_dio\_2, or sf\_dio\_3 line.
- Data is input in 4-bit dual-edge sampling mode through the sf\_dio\_0, sf\_dio\_1, sf\_dio\_2, or sf\_dio\_3 line.

#### 4.2.3.2 SPI Timings

Figure 4-11 shows the SPI timings and related parameters.

**Figure 4-11** SPI output timing



#### NOTE

- The timings are configured in the TIMING\_SPI\_CFG register.
- **tcsh**: **sf\_cs\_n** hold time
- **tcss**: **sf\_clk** nsetup time
- **tshsl**: **sf\_cs\_n** deselect time

#### 4.2.3.3 SPI NAND FLASH Address

Table 4-3 describes the address allocation of the SPI NAND flash. The first and second bytes indicate the column address, whereas the third, fourth, and fifth bytes indicate the row address.

**Table 4-3** Address allocation of the SPI NAND flash

Byte	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
1st Byte	A0	A1	A2	A3	A4	A5	A6	A7
2nd Byte	A8	A9	A10	A11	A12*	A13*	0	0
3rd Byte	A12	A13	A14	A15	A16	A17	A18	A19
4th Byte	A20	A21	A22	A23	A24	A25	A26	A27
5th Byte*	A28*	A29*	0	0	0	0	0	0



#### NOTE

- Bits A0–A11 are configured as the valid column address when the page size is 2 KB. Bits A0–A12 are configured as the valid column address when the page size is 4 KB.
- Bit A12 (2 KB page size) and bit A13 (4 KB page size) indicate the plane address only for Micron flash memories. Other vendors do not have the concept of plane address.
- Whether bits A28 and A29 are required depends on the flash memory. If the two bits are not used, the outputs are 0.

When the read and write commands of the SPI NAND flash are issued, the column and row addresses are configured based on the following operations:

- For the write operation, the column address is configured during the loading process and the row address is configured during the programming process.
- For the read operation, the row address is configured when the page is read to cache and the column address is configured during the read operation.

In internal DMA mode, the address command is issued by the FMC.[FMC\\_ADDR](#) and[FMC\\_ADDRH](#) are configured by software based on the operation address. The configuration values of[FMC\\_ADDR](#) and[FMC\\_ADDRH](#) are 1st byte–4th byte and 5th byte respectively.

#### NOTE

- If the page size is 2 KB, bit A12 indicates the plane address. Other meanings of bit A12 are not supported; otherwise, the results of the read and write operations will be affected.
- If the page size is 4 KB, bit A13 indicates the plane address. Other meanings of bit A13 are not supported; otherwise, the results of the read and write operations will be affected.

### 4.2.3.4 Boot Function

The FMC is in boot mode by default. The chip can boot by reading data directly from the flash memory. The CPU can directly read the data stored in the address ranging from 0x00\_0000 to 0x0F\_FFFF. The size of the entire address space is 1 MB.

### SPI NOR Flash

The address space of the SPI NOR flash is consecutive. The 1 MB boot data is directly mapped to address space 0x00\_0000 to 0x0F\_FFFF of the SPI NOR flash.

Before booting, when the CPU read the data from the boot space for the first time, the FMC obtains the WIP of the flash by running the **Read Status** command. If the WIP is 1, the FMC continues to read data from the boot space until the return value of WIP is 0. At this time, the FM starts to send read operation to the flash.



## SPI NAND Flash

For the SPI NAND flash, the address space is inconsecutive and bad blocks may exist. As a result, the 1 MB boot data cannot be directly mapped to the flash. To implement the boot operation, page\_size, and block\_size are also required for address decoding.

The FMC supports the adaptive boot function, that is, it can automatically adapt to ecc\_type, page\_size, and block\_size of the flash based on the block 0 data. The FMC requires that physical block 0 be a good block, and it can automatically skip other bad blocks.

During booting, the FMC automatically skips bad blocks and searches for good blocks to read boot information. A maximum of four consecutive bad blocks can be skipped at a time. If five consecutive physical bad blocks are encountered, the boot fails. During the boot process, several bad-block skip processes are allowed. A bad-block-skip process ends when a good block is encountered, and a new bad-block-skip process starts when a bad block is encountered.

The FMC logic determines whether a block is a bad block based on the following conditions:

- The bad block flag bits of the first page and last page of the current physical block are 0xff.
- The empty block flag bits of the first page and last page of the current physical block are 0x00.
- The ECC units where the OOB data is located in the first page and last page of the current physical block have no uncorrectable errors.

The FMC logic judges a block as a good block only when all the preceding three conditions are met. Otherwise, the block is judged as a bad block.

### 4.2.3.5 Operations in Register Mode

The registers related to the operation command and address are configured by using software. Then the corresponding command is issued by configuring the [FMC\\_OPRegister](#). The FMC issues a command to the flash memory according to the value configured by software. If data needs to be transferred to the flash memory, the internal buffer is used.

Operations such as reading ID, setting feature, and erasing are performed in this mode.

In register mode, all the operations related to the flash memory can be combined, and commands related to the address and data transfer can be issued separately.

### 4.2.3.6 Operations in Internal DMA Mode

The FMC can perform read and write operations in internal DMA mode for improving the access speed. In internal DMA mode, the FMC can directly access the DDR through the bus.



- DMA write operation: For the SPI NOR flash, data with any length can be transferred from any address of the DDR and written to any address of the flash memory. For the SPI NAND flash, the write operation is performed only by page.
- DMA read operation: For the SPI NOR flash, data with any length can be transferred from any address of the flash memory and written to any address of the DDR. For the SPI NAND flash, the entire page is read or only the OOB is read.
- Only-OOB read operation: When the software requires only software management information such as the bad block flag and empty block flag, only the control information needs to be read. In this case, only the OOB is read in DMA mode.

#### 4.2.3.7 ECC

For the SPI NAND flash, the FMC supports ECC. Three ECC modes are supported, including 8-bit/1 KB ECC mode, 16-bit/1 KB ECC mode, and 24-bit/1 KB ECC mode, ECC mode. For example, in 8-bit/1 KB ECC mode, errors occur in at most 8 bits of the 1 KB\* data can be corrected.

##### NOTE

\* A 1 KB error correction unit contains the ECC protection data, OOB information, and ECC code. The error correction algorithm is calculated based on error correction units. It can be considered that the DATA and OOB make up the data for ECC calculation. If the page size is 2 KB, the data size of each error correction unit is  $(\text{DATA} + \text{OOB})/2$ . If the page size is 4 KB, the data size of each error correction unit is  $(\text{DATA} + \text{OOB})/4$ . Therefore, the actual size of an error correction unit is greater than 1 KB.

- The OOB is part of the software management information. For details, see the (BB + CTRL) part in the data structure.
- DATA indicates the real data. DATA is 2048 bytes when the page size is 2 KB and is 4096 bytes when the page size is 4 KB.

When the maximum error correction capability is exceeded, the uncorrectable error interrupt is reported. The FMC supports the alarm interrupt. When the number of error bits during one error correction operation is greater than or equal to the configured error threshold ([FMC\\_ERR\\_THD](#) [fmc\_err\_thd]), an error alarm interrupt is reported. If error bits occur in one or more error correction units and the number of error bits in each error correction unit is less than the error threshold and uncorrectable error value, the correctable error interrupt flag [FMC\\_INT](#) [err\_val\_int] is changed to 1.

When the data on a page is read, if uncorrectable errors occur in one error correction unit, the uncorrectable error interrupt is reported. If an error alarm interrupt is reported for an error correction unit, the error correction alarm status is reported. If the number of error correction bits is less than the error threshold, the correctable error interrupt status is reported.



The FMC does not support ECC check for the SPI NOR flash. If [FMC\\_CFG](#) [ecc\_type] is 0, the SPI NAND flash works in non-ECC check mode and the FMC transfers data directly from the flash without data structure processing.

#### 4.2.3.8 Timeout

The FMC provides the timeout mechanism for the DMA write operation. When waiting for the completion of the DMA write operation, the FMC continuously sends the **get feature** command to query whether the write operation is complete. For the SPI NOR flash, the operation is complete when the SPI NOR flash sends the **RDSR** command. If the FMC wait time exceeds the configured value of [FMC\\_TIMEOUT\\_WR](#) [timeout\_wr], the DMA write operation is ended and the [FMC\\_INT](#) [op\_fail] interrupt is reported.

### 4.2.4 Working Process

#### 4.2.4.1 Initialization Process

The initialization process is as follows:

- Step 1** (Optional; when the timing parameters need to be adjusted) Configure [TIMING\\_SPI\\_CFG](#) (for the SPI NAND flash or SPI NOR flash) based on the flash memory.
- Step 2** Configure the interface type, ecc\_type and page size of the flash memory in the FMC configuration register ([FMC\\_CFG](#)) according to the manual of the connected flash.
- Step 3** Switch the address mode of the SPI NOR flash to the 4-byte address mode if the default address mode is 3-byte address mode. For details, see section "[4.2.4.4 Process of Changing the Address Mode of the SPI NOR Flash](#)."
- Step 4** Issue operations based on the operation configuration registers.

----End

#### 4.2.4.2 FMC\_OP Operation Process (Register Operation Mode)

To perform operations on the registers of flash memories, for example, to read the IDs or configure flash registers, use FMC\_OP. Specifically, perform the following steps:

- Step 1** Set ecc\_type of the [FMC\\_CFG](#) register to 0.
- Step 2** Write the expected operation data from the buffer access start address for a register write operation (for example, configure the flash memory configuration register).
- Step 3** Configure the operation command, operation address, and number of data segments to be read or written as required in [FMC\\_CMD](#), [FMC\\_ADDRL](#), and [FMC\\_DATA\\_NUM](#) respectively.



- Step 4** Configure [FMC\\_OP\\_CFG](#) based on the flash commands that are issued by configuring [FMC\\_OP](#).
- Step 5** Configure [FMC\\_OP](#) to issue commands. For details about the configuration value, see the description of [FMC\\_OP](#).
- Step 6** Check whether the operation is complete by querying [FMC\\_OP\[reg\\_op\\_start\]](#) in query mode and [FMC\\_INT\[op\\_done\\_int\]](#) in interrupt mode. If [FMC\\_OP\[reg\\_op\\_start\]](#) is 0 or [FMC\\_INT\[op\\_done\\_int\]](#) is detected, the operation is complete.
- Step 7** Read the value of the flash register from the buffer that stores the read results in [Step 6](#).
- End

#### 4.2.4.3 Reading the Flash Status Register

To read the flash status register, perform the following steps:

- Step 1** Set [FMC\\_OP \[read\\_status\\_en\]](#) and [FMC\\_OP \[reg\\_op\\_start\]](#) to 1 to issue the operation.
- Step 2** Store the read status result in [FMC\\_FLASH\\_INFO](#).
- End

#### 4.2.4.4 Process of Changing the Address Mode of the SPI NOR Flash

The SPI NOR flash supports the 3-byte and 4-byte address modes. You can specify the default address mode by pulling up or pulling down the corresponding pin, and dynamically change the address mode by configuring registers after the chip boots.

If the default address mode is the 3-byte address mode and the address mode of the flash memory is 4-byte address mode, perform the following steps to change the address mode after the chip boots:

- Step 1** Ensure that the operations on the SPI NOR flash are complete.
- Step 2** Configure the related registers to issue the command for changing the address mode of the flash memory to 4-byte address mode in register mode according to the flash memory requirements.
- Step 3** Set [FMC\\_CFG \[spi\\_nor\\_addr\\_mode\]](#) to 4B to change the SPI NOR flash from 3-byte address mode to 4-byte address mode.
- End

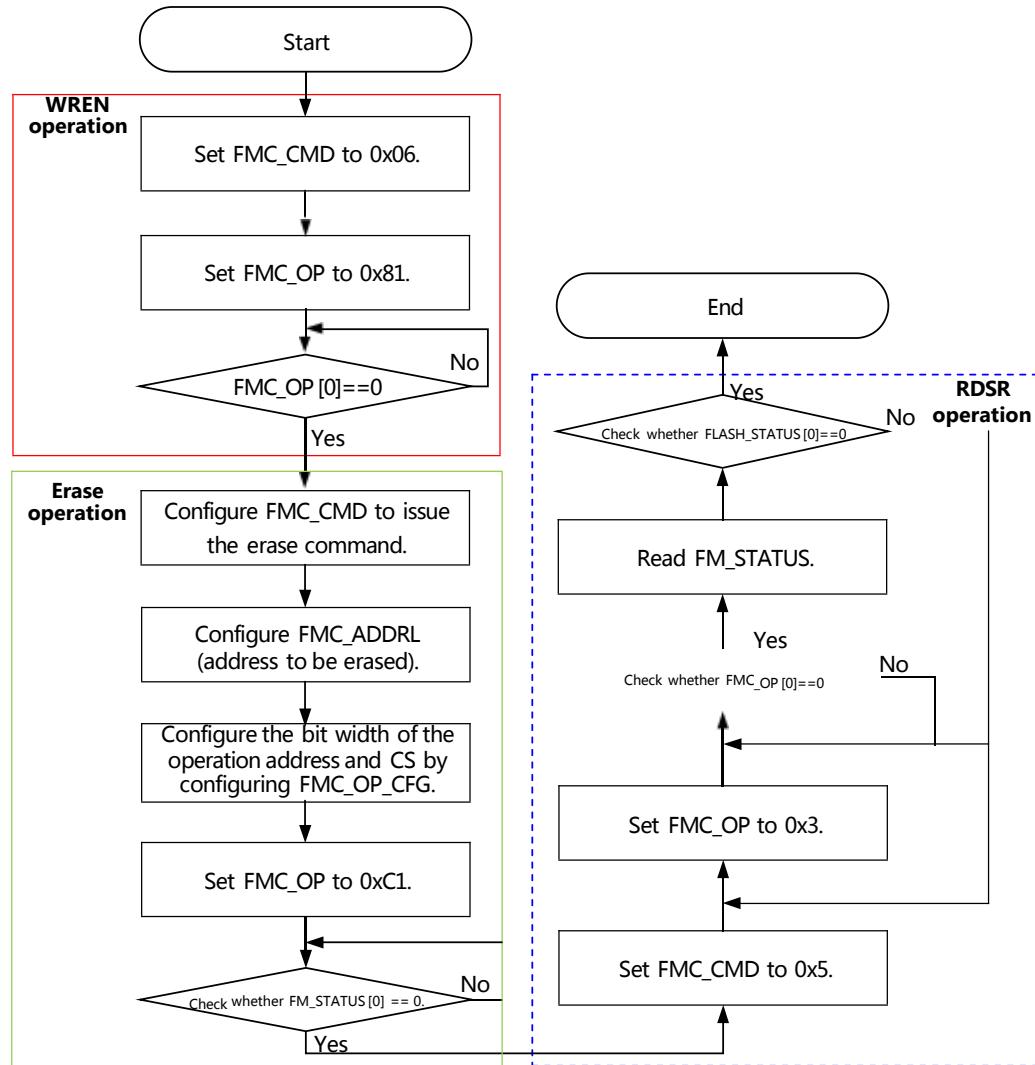
#### NOTE

For details about the command for changing the address mode of the SPI NOR flash, see the related flash manual.

#### 4.2.4.5 Erase Operation Process (SPI NAND flash and SPI NOR flash)

The data in the flash memory must be erased before the program operation is performed. Note that the write enable (WREN) operation must be performed before the erase operation. Figure 4-12 shows an example of erase operation process in query mode.

Figure 4-12 Example of erase operation process





#### NOTE

- Figure 4-12 shows the erase operation process in query mode. If the interrupt mode is used, the operation of querying [FMC\\_OP \[reg\\_op\\_start\]](#) is changed to interrupt handling. If [FMC\\_INT\[op\\_done\\_init\]](#) is detected, the operation is complete.
- During the erase operation, configure the erase instruction and operation address according to the manual of the corresponding flash.

### 4.2.4.6 Process of Reading Data in Internal DMA Mode (FMC\_OP\_CTRL Read Operation)

To read data in internal DMA mode, perform the following steps:

- Step 1** Set [FMC\\_CFG \[op\\_mode\]](#) to 1 to ensure that the FMC is in normal mode.
- Step 2** Configure the operation address of the flash memory by configuring [FMC\\_ADDR](#) and [FMC\\_ADDRH](#). For the SPI NOR flash, only [FMC\\_ADDR](#) needs to be configured.

#### NOTICE

For the SPI NOR flash, the number of address cycles during the DMA operation is determined by [FMC\\_CFG\[spi\\_nor\\_addr\\_mode\]](#). The 3-byte and 4-byte address modes are supported. For the SPI NAND flash, the FMC uses the 5-byte address mode by default and the address mode cannot be configured.

- Step 3** Configure the start address for storing data in the DDR by configuring [FMC\\_DMA\\_SADDR\\_OOB](#).
  - For the SPI NOR flash, [FMC\\_DMA\\_SADDR\\_OOB](#) does not need to be configured.
  - For the SPI NAND flash, if only the OOB is read, only [FMC\\_DMA\\_SADDR\\_OOB](#) need to be configured.

#### NOTICE

- ☞ For the SPI NAND flash, the DDR address must be 4-byte aligned.
- ☞ For the SPI NAND flash in ECC0 mode, the length configured in [FMC\\_DMA\\_LEN](#) must be 4-byte aligned.

- Step 4** Configure [FMC\\_DMA\\_LEN](#). For the SPI NOR flash, [FMC\\_DMA\\_LEN](#) needs to be set to the length of the read data; for the SPI NAND flash in ECC0 mode, [FMC\\_DMA\\_LEN](#) needs to be set to the length of data in the spare area. This register does not need to be configured for other operations.
- Step 5** Configure [FMC\\_OP\\_CFG](#) based on the requirements of the issued read command.



- For the SPI NAND flash and SPI NOR flash, [FMC\\_OP\\_CFG\[dummy\\_num\]](#) and [FMC\\_OP\\_CFG\[mem\\_if\\_type\]](#) need to be configured based on the number of dummy cycles in the read timings of the SPI flash and the SPI type.
- Configure [FMC\\_OP\\_CFG\[fm\\_cs\]](#) to select the CS to be operated.

**Step 6** Set [FMC\\_OP\\_CTRL\[dma\\_op\\_ready\]](#) to 1 to issue the flash read command.

- Set [FMC\\_OP\\_CTRL\[rw\\_op\]](#) to 0 to select the DMA read operation.
- For the SPI NAND flash, if only the OOB is read, [FMC\\_OP\\_CTRL\[rd\\_op\\_sel\]](#) needs to be configured.
- Configure [FMC\\_OP\\_CTRL\[rd\\_opcode\]](#) based on the instruction of the flash read operation.

**Step 7** Check whether the read operation is complete by querying [FMC\\_OP\\_CTRL\[bit\[0\]\]](#) in query mode and [FMC\\_INT\[op\\_done\\_int\]](#) in interrupt mode. If [FMC\\_OP\\_CTRL\[bit\[0\]\]](#) is 0 or [FMC\\_INT\[op\\_done\\_int\]](#) is 1, the read operation is complete, and data is written to the DDR.

----End

#### 4.2.4.7 Process of Writing Data in Internal DMA Mode ([FMC\\_OP\\_CTRL](#) Write Operation)

To write data in internal DMA mode, perform the following steps:

**Step 1** Set [FMC\\_CFG\[op\\_mode\]](#) to 1 to ensure that the FMC is in normal mode.

**Step 2** Configure the operation address of the flash memory by configuring [FMC\\_ADDR](#) and [FMC\\_ADDRH](#). For the SPI NOR flash, only [FMC\\_ADDR](#) needs to be configured.

##### NOTICE

For the SPI NOR flash, the number of address cycles during the DMA operation is determined by [FMC\\_CFG\[spi\\_nor\\_addr\\_mode\]](#). The 3-byte and 4-byte address modes are supported. For the SPI NAND flash, the FMC uses the 5-byte address mode by default and the address mode cannot be configured.

**Step 3** Configure the start address for transferring data from the DDR by configuring [FMC\\_DMA\\_SADDR\\_D0](#) and [FMC\\_DMA\\_SADDR\\_OOB](#). For the SPI NOR flash, [FMC\\_DMA\\_SADDR\\_OOB](#) does not need to be configured.



## NOTICE

- ☞ For the SPI NAND flash, the DDR address must be 4-byte aligned.
- ☞ For the SPI NAND flash in ECC0 mode, the length configured in [FMC\\_DMA\\_LEN](#) must be 4-byte aligned.

- Step 4** Configure [FMC\\_DMA\\_LEN](#). For the SPI NOR flash, [FMC\\_DMA\\_LEN](#) needs to be set to the length of the read data; for the SPI NAND flash in ECC0 mode, [FMC\\_DMA\\_LEN](#) needs to be set to the length of data in the spare area. This register does not need to be configured for other operations.
- Step 5** Configure [FMC\\_OP\\_CFG](#) based on the requirements of the issued write command.
- For the SPI NAND flash and SPI NOR Flash, configure [FMC\\_OP\\_CFG](#) [mem\_if\_type] based on the SPI required for the write operation.
  - Configure [FMC\\_OP\\_CFG](#) [fm\_cs] to select the CS to be operated.
- Step 6** Configure [FMC\\_OP\\_CTRL](#) to issue corresponding commands.
- Set [FMC\\_OP\\_CTRL](#) [rw\_op] to 1 to select the DMA write operation.
  - For the SPI NAND flash and SPI NOR flash, configure [FMC\\_OP\\_CTRL](#) [wr\_opcode] based on the instruction of the flash program operation.
- Step 7** Check whether the write operation is complete by querying [FMC\\_OP\\_CTRL](#) bit[0] in query mode and [FMC\\_INT](#) [op\_done\_int] in interrupt mode. If [FMC\\_OP\\_CTRL](#) bit[0] is 0 or [FMC\\_INT](#) [op\_done\_int] is 1, the write operation is complete, and data is written to the flash memory.

----End

### 4.2.4.8 Notes

Note the following:

- You must reset the SPI NAND flash before use or after exceptions occur.
- You are advised not to configure registers when [FMC\\_OP\\_CTRL](#) [dma\_op\_ready] or [FMC\\_OP](#) [reg\_op\_start] is 1, indicating that the controller is performing an operation. Otherwise, the operation may become abnormal.
- Ensure that software configurations follow correct and appropriate rules; otherwise, the FMC may be suspended.

### 4.2.5 Data Structures (SPI NAND Flash)

There are two types of FMC data. One is the original user data, the other is the data converted by the FMC and stored in the SPI NAND flash in non-ECC0 mode. In non-ECC0 mode, the original user data and data converted by the FMC have



fixed data structures as well as data member length and position. In ECC0 mode, the user data is written or read without being changed by the FMC.

In ECC0 mode, the FMC transparently transmits data. During the write operation, the FMC directly writes the data in the buffer to the flash; during the read operation, the FMC directly writes the data read from the flash to the buffer.

In non-ECC0 mode, when the flash is written, the user data processing by the FMC involves ECC code generation and data reconstitution. The FMC implements ECC encoding in the internal buffer, generates the ECC code, and reconstitutes buffer data, and writes data to the flash. When the flash is read, the data processing by the FMC involves flash data reconstitution and ECC. The FMC reconstitutes data read from the flash, writes data to the internal buffer, and implements ECC in the buffer.

In non-ECC0 mode, data in the buffer and data in the flash have different structures. [Table 4-4](#) describes the length of each data segment in various non-ECC0 modes.

**Table 4-4** Data structure length in various non-ECC0 modes

ECC <sup>a</sup> (bit)	ecc_len <sup>b</sup> (Byte)	page_size(Byte)	oob_len <sup>c</sup> (Byte)	sec_len <sup>d</sup> (Byte)
4/512B (8/1K)	14	2048	30+2	1040
		4096		1032
8/512B (16/1K)	28	2048	6+2	1028
		4096		
24/1K	42	2048	30+2	1040
		4096		1032

#### NOTE

a: The 4-bit/512 bytes mode is equivalent to the 8-bit/1 KB mode, and the 8-bit/512 bytes mode is equivalent to the 16-bit/1 KB mode.

b: ecc\_len indicates the length of ECC code generated by each error correction unit.

c: oob\_len indicates the length of the redundant area visible to the upper-layer software, which consists of the CTRL and BB.

d: sec\_len indicates the length of the data area in each error correction unit. sec\_len is calculated as follows:  $sec\_len = 1024 + (oob\_len \times 1024) / Page\_size$

e: The length of each ECC error correction unit is 1 KB. 1 KB means the 1 KB level but not exactly 1024 bytes. For example, the length of the error correction unit (sec\_len) ranges from 1026 bytes to 1040 bytes, and the lengths are all represented as 1 KB.

The following describes each data segment by taking the write operation as an example:



When the FMC writes data to the flash, data is consecutively stored in the format of data+ECC. However, special processing is required for the OOB data.

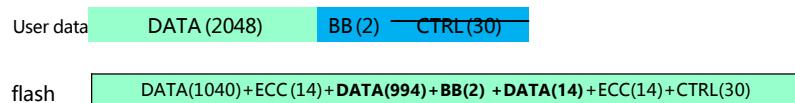
- The BB identity needs to be stored in the position of **page\_size** on the flash page. That is, when the page size is 2 KB or 4 KB, the BB data is stored in the first two bytes of the 2048 or 4096 bytes. In this way, the storage of the original data or ECC data is affected.
- The CTRL data needs to be stored at the end of the valid data. Therefore, the storage format of the last data segment in the flash is data (excluding BB data and CTRL data)+ECC+CTRL.
- EB is the flag for marking the empty block and is usually stored in the last 2 bytes of the CTRL data.

#### 4.2.5.1 4-Bit ECC Mode (8-Bit/1 KB Error Correction Performance)

##### 2 KB page\_size

When the page size is 2 KB, the size of the redundant area available to software is 32 bytes. [Figure 4-13](#) shows the structure of data in the buffer and flash.

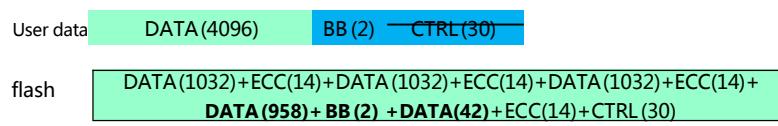
**Figure 4-13** Data structure when the page size is 2 KB in 4-bit ECC mode



##### 4 KB page\_size

When the page size is 4 KB, the size of the redundant area available to software is 32 bytes. [Figure 4-14](#) shows the structure of data in the buffer and flash.

**Figure 4-14** Data structure when the page size is 4 KB in 4-bit ECC mode



#### 4.2.5.2 8-Bit ECC Mode (16-Bit/1 KB Error Correction Performance)

##### 2 KB page\_size

When the page size is 2 KB, the size of the redundant area available to software is 8 bytes. [Figure 4-15](#) shows the structure of data in the buffer and flash.



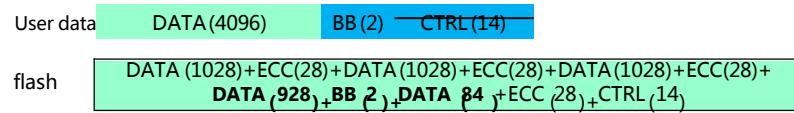
**Figure 4-15** Data structure when the page size is 2 KB in 8-bit ECC mode



## 4 KB page\_size

When the page size is 4 KB, the size of the redundant area available to software is 16 bytes. [Figure 4-16](#) shows the structure of data in the buffer and flash.

**Figure 4-16** Data structure when the page size is 4 KB in 8-bit ECC mode

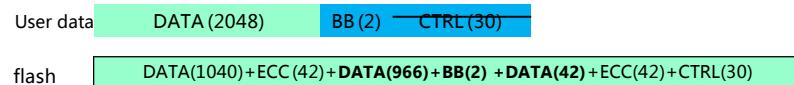


### 4.2.5.3 24-Bit ECC Mode (24-Bit/1 KB Error Correction Performance)

#### 2 KB page-size

When the page size is 2 KB, the size of the redundant area available to software is 32 bytes. [Figure 4-17](#) shows the structure of data in the buffer and flash.

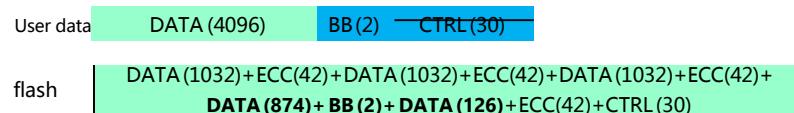
**Figure 4-17** Data structure when the page size is 2 KB in 24-bit ECC mode



#### 4 KB page\_size

When the page size is 4 KB, the size of the redundant area available to software is 32 bytes. [Figure 4-18](#) shows the structure of data in the buffer and flash.

**Figure 4-18** Data structure when the page size is 4 KB in 24-bit ECC mode



### 4.2.6 ECC Mode Selection

The ECC IP used by the controller processes data based on 1 KB data block. Therefore, the ECC performance of the controller is described in the format of  $n$



bits/1 KB, for example, 8 bits/1 KB or 24 bits/1 KB. To ensure reliability of the flash memory, the recommended ECC performance of 1 bit/512 bytes, 4 bits/512 bytes, or 8 bits/512 bytes is provided in the flash data sheet. The ECC performance of 8 bits/1 KB is equivalent to that of 4 bits/512 bytes. When you select the ECC mode for the controller, note the following:

- The priority of the ECC performance of the controller is higher than or equal to that of the recommended ECC performance.  
If the recommended ECC performance is 1 bit/512 bytes or 4 bits/512 bytes, the controller can select the ECC mode with the performance of 8 bits/1 KB.
- The page size of the flash memory must be greater than or equal to the required storage size of the ECC mode for the controller.  
If the page size of the flash memory is (2 KB + 64 bytes), the controller cannot select the ECC mode with the performance of 24 bits/1 KB because the required storage size of this ECC mode is (2 KB + 116 bytes). For details about the required storage size for the ECC mode, see section "[4.2.5 Data Structures \(SPI NAND Flash\)](#)."

## 4.2.7 FMC Registers

[Table 4-5](#) describes FMC registers.

**Table 4-5** Summary of FMC registers (base address: 0x1000\_0000)

Offset Address	Register	Description	Page
0x0000	FMC_CFG	Flash configuration register	<a href="#">4-29</a>
0x0004	GLOBAL_CFG	Global configuration register	<a href="#">4-30</a>
0x0008	TIMING_SPI_CFG	SPI timing configuration register	<a href="#">4-32</a>
0x0018	FMC_INT	Interrupt status register	<a href="#">4-32</a>
0x001C	FMC_INT_EN	Interrupt enable register	<a href="#">4-34</a>
0x0020	FMC_INT_CLR	Interrupt clear register	<a href="#">4-35</a>
0x0024	FMC_CMD	Command word configuration register	<a href="#">4-35</a>
0x0028	FMC_ADDRH	Upper-byte flash address configuration register	<a href="#">4-36</a>
0x002C	FMC_ADDRL	Lower-4-byte flash address configuration register	<a href="#">4-36</a>
0x0030	FMC_OP_CFG	Operation configuration register	<a href="#">4-36</a>
0x0034	SPI_OP_ADDR	Operation address configuration register	<a href="#">4-37</a>



Offset Address	Register	Description	Page
0x0038	FMC_DATA_NUM	Data length register	<a href="#">4-37</a>
0x003C	FMC_OP	Operation register	<a href="#">4-38</a>
0x0040	FMC_DMA_LEN	DMA operation length register	<a href="#">4-39</a>
0x0048	FMC_DMA_AHB_CTL	DMA AHB bus control register	<a href="#">4-40</a>
0x004C	FMC_DMA_SADDR_D0	DDR SDRAM start address register for DMA operations	<a href="#">4-40</a>
0x005C	FMC_DMA_SADDR_OOB	DDR SDRAM OOB information storage start address register for DMA operations	<a href="#">4-40</a>
0x0068	FMC_OP_CTRL	DMA operation control register	<a href="#">4-41</a>
0x006C	FMC_TIMEOUT_WR	Write operation timeout register	<a href="#">4-42</a>
0x0070	FMC_OP_PARA	OP operation parameter selection register	<a href="#">4-42</a>
0x0074	FMC_BOOT_SET	Boot setting register	<a href="#">4-42</a>
0x0078	FMC_LP_CTRL	Low-power control register	<a href="#">4-43</a>
0x00A8	FMC_ERR_THD	ECC alarm threshold register	<a href="#">4-43</a>
0x00AC	FMC_FLASH_INFO	Flash status register	<a href="#">4-44</a>
0x00BC	FMC_VERSION	Version register	<a href="#">4-45</a>
0x00C0	FMC_ERR_NUM0_BUFO	SPI NAND Flash error correction information 0 statistics register for the first buffer operation	<a href="#">4-45</a>
0x00D0	FMC_ERR_ALARM_A_DDRH	Upper-byte ECC alarm flash address register	<a href="#">4-45</a>
0x00D4	FMC_ERR_ALARM_A_DDRL	Lower-byte ECC alarm flash address register	<a href="#">4-46</a>
0x00D8	FMC_ECC_INVALID_ADDRH	Upper-byte ECC uncorrectable address register	<a href="#">4-46</a>
0x00DC	FMC_ECC_INVALID_ADDRL	Lower-4-byte ECC uncorrectable address register	<a href="#">4-46</a>
0x0037C	FMC_BOOT_CMD	Boot read command configuration register	<a href="#">4-46</a>



## 4.2.8 FMC Register Description

### FMC\_CFG

FMC\_CFG is the flash configuration register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_1820

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12:11]	RW	spi_nand_plane_en	Plane bit function enable. The plane address bit information is sent when the address is sent, which is valid only in SPI NAND mode. 11: enabled Other values: disabled	0x3
[10]	RW	spi_nor_addr_mode	SPI address mode (valid only for the SPI NOR flash) 0: 3-byte address mode (default) 1: 4-byte address mode The reset value depends on the pin.	0x0
[9:8]	RW	block_size	Block size of the SPI NAND flash 00: 64 page 01: 128 page Other values: reserved	0x0
[7:5]	RW	ecc_type	ECC type of the controller 000: no ECC 001: 8-bit ECC 010: 16-bit ECC 011: 24-bit ECC Other values: reserved	0x1
[4:3]	RW	page_size	Page size of the SPI NAND flash 00: 2KB page_size 01: 4KB page_size Other values: reserved	0x0
[2:1]	RW	flash_sel	Flash type select 00: SPI NOR FLASH	0x0



Bits	Access	Name	Description	Reset
			01: SPI NAND FLASH Other values: reserved The reset value depends on the pin.	
[0]	RW	op_mode	FMC operation mode 0: boot mode 1: normal mode	0x0

## GLOBAL\_CFG

GLOBAL\_CFG is the global configuration register.

Offset Address: 0x0004 Total Reset Value: 0x0001\_00C4

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	Reserved	0x0000
[20]	RW	ddr_mode_byp	Bypass function control for the SPI NAND DDR, ensuring the correctness of all previous functions 1: The bypass function is enabled. 0: The bypass function is disabled.   <b>NOTE</b> This bit field applies only to the SPI NAND DDR.	0x0
[19:16]	-	reserved	Reserved	0x1
[15:12]	RW	sample_point	Sampling point select based on the delay parameter of the flash memory in DDR mode. The frequency of the sampling clock is four times that of the interface read/write clock. The delay is the lowest when the field value is 0x0, and highest when the field value is 0xB. 0x0: The sampling point is 2 sampling clock cycles after the valid edge. 0x1: The sampling point is 2.5 sampling clock cycles after the valid edge. 0x2: The sampling point is 3 sampling clock cycles after the valid edge. 0x3: The sampling point is 3.5 sampling clock cycles after the valid edge.	0x0



Bits	Access	Name	Description	Reset
			0x4: The sampling point is 4 sampling clock cycles after the valid edge. 0x5: The sampling point is 4.5 sampling clock cycles after the valid edge. 0x6: The sampling point is 5 sampling clock cycles after the valid edge. 0x7: The sampling point is 5.5 sampling clock cycles after the valid edge. 0x8: The sampling point is 6 sampling clock cycles after the valid edge. 0x9: The sampling point is 6.5 sampling clock cycles after the valid edge. 0xA: The sampling point is 7 sampling clock cycles after the valid edge. 0xB: The sampling point is 7.5 sampling clock cycles after the valid edge. Other values: reserved	
[11]	RW	ddr_mode	DDR mode enable 0: normal SDR mode 1: DDR mode	0x0
[10:7]	-	reserved	Reserved	0x1
[6]	RW	wp_en	Write protection enable for the WP pin. When this bit is enabled, the chip outputs 0 to the WP pin. 0: disabled 1: enabled	0x1
[5:3]	RW	rd_delay	Number of delayed cycles for reading data from the SPI flash (in SDR mode) 000 (default): no delay 001: 0.5 cycle 010: 1 cycle 011: 1.5 cycles 100: 2 cycles 101: 2.5 cycles 110: 3 cycles 111: 3.5 cycles	0x0
[2]	-	reserved	Reserved	0x1



Bits	Access	Name	Description	Reset
[1:0]	-	reserved	Reserved	0x0

## TIMING\_SPI\_CFG

TIMING\_SPI\_CFG is the SPI timing configuration register.

Offset Address: 0x0008 Total Reset Value: 0x0000\_006F

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:4]	RW	tcss	CS setup time 000–111: ( $n + 1$ ) interface clock cycles ( $n = 0, 1, 2, \dots, 7$ )	0x6
[3:0]	RW	tshsl	CS deselect time. It is equal to the interval between two flash operations. 0000–1111: ( $n + 1$ ) interface clock cycles ( $n = 0, 1, 2, \dots, 15$ )	0xF

## FMC\_INT

FMC\_INT is the interrupt status register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7]	RO	ahb_op_int	CPU read/write internal buffer interrupt enable when the FMC is reading/writing data from/to the flash memory 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	-	reserved	Reserved	0x0
[5]	RO	dma_err_int	DMA transfer bus error interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	err_alarm_int	ECC alarm interrupt. An interrupt is	0x0



Bits	Access	Name	Description	Reset
			generated when the number of error bits reaches the preset threshold. 0: No interrupt is generated. 1: An interrupt is generated.	
[3]	RO	err_inval_int	Uncorrectable ECC error interrupt In 8-bit ECC mode, if errors occur in eight or more bits of the checked 1024-byte data, an interrupt is generated. In 16-bit ECC mode, if errors occur in 16 or more bits of the checked 1024-byte data, an interrupt is generated. In 24-bit ECC mode, if errors occur in 24 or more bits of the checked 1024-byte data, an interrupt is generated. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	RO	err_val_int	Correctable ECC error interrupt In 8-bit ECC mode, if errors occur in one to 8 bits of the checked 1024-byte data, an interrupt is generated. In 16-bit ECC mode, if errors occur in one to 16 bits of the checked 1024-byte data, an interrupt is generated. In 24-bit ECC mode, if errors occur in one to 24 bits of the checked 1024-byte data, an interrupt is generated. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	op_fail_int	Programming failure interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	op_done_int	Current controller operation completion interrupt. This bit is automatically cleared when the operation register is written. 0: No interrupt is generated. 1: An interrupt is generated.	0x0



## FMC\_INT\_EN

FMC\_INT\_EN is the interrupt enable register.

Offset Address: 0x001C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7]	RW	ahb_op_int_en	CPU read/write internal buffer error interrupt enable when the FMC is reading/writing data from/to the flash memory 0: disabled 1: enabled	0x0
[6]	-	reserved	Reserved	0x0
[5]	RW	dma_err_int_en	DMA transfer bus error interrupt enable 0: disabled 1: enabled	0x0
[4]	RW	err_alarm_int_en	ECC alarm interrupt enable. An interrupt is generated when the number of error bits reaches the threshold 0: disabled 1: enabled	0x0
[3]	RW	err_inval_int_en	ECC uncorrectable error interrupt enable 0: disabled 1: enabled	0x0
[2]	RW	err_val_int_en	ECC correctable error interrupt enable 0: disabled 1: enabled	0x0
[1]	RW	op_fail_int_en	Programming operation failure interrupt enable 0: disabled 1: enabled	0x0
[0]	RW	op_done_int_en	Current operation completion interrupt enable of the FMC 0: disabled 1: enabled	0x0



## FMC\_INT\_CLR

FMC\_INT\_CLR is the interrupt clear register.

Offset Address: 0x0020 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7]	WO	ahb_op_int_clr	cache_page_done interrupt clear. Writing 1 clears the interrupt.	0x0
[6]	-	reserved	Reserved	0x0
[5]	WO	dma_err_int_clr	DMA transfer bus error interrupt clear. Writing 1 clears the interrupt.	0x0
[4]	WO	err_alarm_int_clr	err_alarm interrupt clear. Writing 1 clears the interrupt.	0x0
[3]	WO	err_inval_int_clr	err_invalid interrupt clear. Writing 1 clears the interrupt.	0x0
[2]	WO	err_val_int_clr	err_valid interrupt clear. Writing 1 clears the interrupt.	0x0
[1]	WO	op_fail_int_clr	op_fail interrupt clear. Writing 1 clears the interrupt.	0x0
[0]	WO	op_done_int_clr	op_done interrupt clear. Writing 1 clears the interrupt.	0x0

## FMC\_CMD

FMC\_CMD is the command word configuration register.

Offset Address: 0x0024 Total Reset Value: 0x0000\_3000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:8]	-	reserved	Reserved	0x30
[7:0]	RW	cmd1	First command sent to the SPI NOR/NAND flash by the FMC	0x00



## FMC\_ADDRH

FMC\_ADDRH is the upper-byte flash address configuration register.

Offset Address: 0x0028 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:0]	RW	addrh	Upper byte of the operation address for configuring the SPI NAND flash	0x00

## FMC\_ADDRL

FMC\_ADDRL is the lower-4-byte flash address configuration register.

Offset Address: 0x002C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	addrl	Lower 4 bytes of the operation address for configuring the flash (For the SPI NOR flash, configure the flash address.)	0x00000000

## FMC\_OP\_CFG

FMC\_OP\_CFG is the operation configuration register.

Offset Address: 0x0030 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13]	RW	multi_cycle_en	SPI redundancy period enable 0: disabled 1: enabled When the frequency of the output clock exceeds 50 MHz, this bit must be set to 1.	0x0
[12:11]	RW	fm_cs	CS corresponding to the flash to be operated 00: cs0; 01: cs1; Other values: reserved.	0x0



Bits	Access	Name	Description	Reset
[10]	RW	force_cs_en	CS forcible pull-down enable 0: disabled 1: enabled	0x0
[9:7]	RW	mem_if_type	SPI flash interface type select (read operation) 000: Standard SPI 001: Dual-Input/Dual-Output SPI 010: Dual-I/O SPI 011: Quad-Input/Quad-Output SPI 100: Quad-I/O SPI 101-111: reserved	0x0
[6:4]	RW	addr_num	Number of bytes of the address sent to the flash	0x0
[3:0]	RW	dummy_num	Number of bytes to be operated for dummy_en (one byte is equivalent to two clock cycles in 4-wire mode, four clock cycles in 2-wire mode, or eight clock cycles in 1-wire mode)	0x0

## SPI\_OP\_ADDR

SPI\_OP\_ADDR is the operation address configuration register.

Offset Address: 0x0034 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	spi_op_addr	Operation address of the SPI NOR/NAND flash. The operation address of the SPI flash is issued for each operation. This address is different from that described in FMC_ADDRL and FMC_ADDRH.	0x0000_0000

## FMC\_DATA\_NUM

FMC\_DATA\_NUM is the data length register.

Offset Address: 0x0038 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13:0]	RW	op_data_num	Length of data to be processed during one operation. This register needs to be configured when there is data transmission and does not need to be configured for the DMA operations and AHB direct access. This register is valid only in ECC0 mode.	0x0000

## FMC\_OP

FMC\_OP is the operation register.

Offset Address: 0x003C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	Reserved	0x0000000
[8]	RW	dummy_en	Dummy byte transfer enable 0: disabled 1: enabled The dummy byte is sent after the address operation when they are both enabled.	0x0
[7]	RW	cmd1_en	Enable for transmitting command 1 to the flash 0: disabled 1: enabled	0x0
[6]	RW	addr_en	Enable for writing the operation address to the flash 0: disabled 1: enabled	0x0
[5]	RW	write_data_en	Enable for writing data to the flash 0: disabled 1: enabled <b>NOTE</b> read_data_en and write_data_en cannot be 1 at the same time.	0x0
[4:3]	-	reserved	Reserved	0x0



Bits	Access	Name	Description	Reset
[2]	RW	read_data_en	Enable for reading data from the flash 0: disabled 1: enabled <b>NOTE</b> read_data_en and write_data_en cannot be 1 at the same time.	0x0
[1]	RW	read_status_en	Enable for reading the flash status register 0: disabled 1: enabled When this bit is set to 1, the component status register value read by the controller is stored in the <b>fm_status</b> field of <b>FMC_FLASH_INFO</b> (not written to the internal buffer).	0x0
[0]	RWSC	reg_op_start	Controller status when commands are issued by configuring the FMC_OP register 0: The controller is ready. This bit can be set only to 1 when software issues the command, indicating that the logic is enabled. 1: The controller is busy. This bit is automatically set to 0 after the operation is complete, indicating that the logic is complete.	0x0

## FMC\_DMA\_LEN

FMC\_DMA\_LEN is the DMA operation length register.

Offset Address: 0x0040 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:0]	RW	dma_len	Data transfer length during DMA operations (in byte) This field is used to configure the length of data in the spare data when the SPI NAND flash memory uses the ECC0 mode. This field is used to configure the length of	0x00000000



Bits	Access	Name	Description	Reset
			data required for the DMA read/write operation of the SPI NOR flash.	

## FMC\_DMA\_AHB\_CTRL

FMC\_DMA\_AHB\_CTRL is the DMA AHB control register.

Offset Address: 0x0048 Total Reset Value: 0x0000\_0007

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2]	RW	burst16_en	Burst16 enable 0: disabled 1: enabled	0x1
[1]	RW	burst8_en	Burst8 enable 0: disabled 1: enabled	0x1
[0]	RW	burst4_en	Burst4 enable 0: disabled 1: enabled	0x1

## FMC\_DMA\_SADDR\_D0

FMC\_DMA\_SADDR\_D0 is DDR SDRAM start address register for DMA operations.

Offset Address: 0x004C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_mem_saddr_d0	Start address for the FMC DMA to access the DDR SDRAM	0x0000_0000

## FMC\_DMA\_SADDR\_OOB

FMC\_DMA\_SADDR\_OOB is the DDR SDRAM OOB information storage start address register for DMA operations.

Offset Address: 0x005C Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	dma_mem_saddr_oob	DDR SDRAM base address for the OOB area that stores the data to be read or written	0x0000_0000

## FMC\_OP\_CTRL

FMC\_OP\_CTRL is the DMA operation control register.

Offset Address: 0x0068 Total Reset Value: 0x0003\_0200

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:16]	RW	rd_opcode	SPI NAND/SPI NOR flash, DMA read command, non-bus mode (FAST_READ/READ/DUAL_READ)	0x03
[15:8]	RW	wr_opcode	SPI NAND/SPI NOR flash, DMA write command, non-bus mode	0x02
[7:6]	-	reserved	Reserved	0x0
[5:4]	RW	rd_op_sel	Area of data to be read 00: Data on the entire page is read. 01: Only the OOB data is read. Other values: reserved	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RW	rw_op	DMA write/read mode select 0: DMA read mode 1: DMA write mode	0x0
[0]	RWSC	dma_op_ready	Controller status when commands are issued by configuring the FMC_OP_CTRL register 0: The controller is ready. This bit can be set only to 1 when software issues the command, indicating that the logic is enabled. 1: The controller is busy. This bit is automatically set to 0 after the operation is complete, indicating that the logic is complete.	0x0



## FMC\_TIMEOUT\_WR

FMC\_TIMEOUT\_WR is the write operation timeout register.

Offset Address: 0x006C Total Reset Value: 0x00FF\_FFFF

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	timeout_wr	Program operation busy wait time, timeout period. The timeout period is in the unit of one interface clock cycle for the SPI NAND/SPI NOR flash.	0x0000000

## FMC\_OP\_PARA

FMC\_OP\_PARA is the OP operation parameter selection register.

Offset Address: 0x0070 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	rd_oob_only	Only the sector that stores the OOB information is read when the flash read command is issued. 0: The page is read. 1: Only the OOB information is read.	0x0
[0]	-	reserved	Reserved	0x0

## FMC\_BOOT\_SET

FMC\_BOOT\_SET is the boot setting register.

Offset Address: 0x0074 Total Reset Value: 0x0000\_0005

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	boot_otp_cfg	Flash configuration information, that is, the mode in which the <a href="#">FMC_CFG</a> register is configured	0x0



Bits	Access	Name	Description	Reset
			0: adaptive boot mode 1: OTP mode	
[3]	-	reserved	Reserved	0x0
[2]	-	reserved	Reserved	0x1
[1]	RW	boot_quad_mode	Whether to use the 4-wire boot mode for the SPI NAND flash. The reset value depends on the pin. 0: No. The 1-wire boot mode is used. 1: Yes.	0x0
[0]	RW	boot_page0_cfg	Boot mode select (SPI NAND flash) 0: default boot mode 1: boot mode without configuring pins	0x1

## FMC\_LP\_CTRL

FMC\_LP\_CTRL is the low-power control register.

Offset Address: 0x0078 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	mem_gate_en	SRAM clock enable 0: enabled 1: disabled	0x0
[0]	RW	clk_gate_en	Clock gating select. If the clock gating is enabled, the low-power design is used and the module working clocks are disabled by logic. 0: All clocks are enabled. 1: Clocks are disabled according to the low-power design.	0x0

## FMC\_ERR\_THD

FMC\_ERR\_THD is the ECC alarm threshold register.

Offset Address: 0x00A8 Total Reset Value: 0x0000\_00FF



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	fmc_err_thd	ECC alarm threshold. When the number of error bits reaches the threshold, an ECC alarm interrupt is triggered.  <b>NOTE</b> <ul style="list-style-type: none"><li>④ If an uncorrectable ECC error occurs, an uncorrectable interrupt is reported regardless of the register value.</li><li>④ When the register value is set to 0 or 1, an alarm interrupt is reported as long as an error occurs.</li><li>④ When the configured value exceeds the number of correctable error bits, no alarm interrupt is reported regardless of the number of error bits. Only the correctable error interrupt and uncorrectable error interrupt are generated.</li></ul>	0xFF

## FMC\_FLASH\_INFO

FMC\_FLASH\_INFO is the flash status register.

Offset Address: 0x00AC Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	exp_bb_flag	Uncorrected data corresponding to the bad block flag of the current page when the extended operation of reading only the OOB data or reading the page is issued	0x00
[23:16]	RO	bb_flag	Uncorrected data corresponding to the bad block flag of the current page when the only the OOB data or the page is read	0x00
[15:8]	RO	exp_flash_status	Status register value read by the FMC from the flash during extended operations	0x00
[7:0]	RO	flash_status	Status register value read by the FMC from the flash during normal operations. For details about the meanings of bits 7-0, see the flash data sheet.	0x00



## FMC\_VERSION

FMC\_VERSION is the version register.

Offset Address: 0x00BC Total Reset Value: 0x0000\_0100

Bits	Access	Name	Description	Reset
[31:0]	RO	version	FMC V100	0x0000_0100

## FMC\_ERR\_NUM0\_BUFO

FMC\_ERR\_NUM0\_BUFO is the SPI NAND flash error correction information 0 statistics register for the first buffer operation.

Offset Address: 0x00C0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	err_num0_buf0	Error bit count for the first 4 KB data during the first buffer operation of the NAND flash with the page size of 2 KB, 4 KB, 8 KB, or 16 KB  bit[31:24]: number of error bits in the fourth 1 KB data segment bit[23:16]: number of error bits in the third 1 KB data segment bit[15:8]: number of error bits in the second 1 KB data segment bit[7:0]: number of error bits in the first 1 KB data segment	0x0000_0000

## FMC\_ERR\_ALARM\_ADDRH

FMC\_ERR\_ALARM\_ADDRH is the upper-byte ECC alarm flash address register.

Offset Address: 0x00D0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x00000000
[7:0]	RO	fmc_err_alarm_addrh	Upper bytes of the operation address of the flash that triggers the last interrupt when an ECC alarm interrupt is generated	0x00



## FMC\_ERR\_ALARM\_ADDR

FMC\_ERR\_ALARM\_ADDR is the lower-byte ECC alarm flash address register.

Offset Address: 0x00D4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	fmc_err_alarm_addr	Lower bytes of the operation address of the flash that triggers the last interrupt when an ECC alarm interrupt is generated	0x0000_0000

## FMC\_ECC\_INVALID\_ADDRH

FMC\_ECC\_INVALID\_ADDRH is the upper-byte ECC uncorrectable address register.

Offset Address: 0x00D8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:0]	RO	fmc_ecc_invalid_addrh	Upper byte of the operation address of the flash that triggers the last interrupt when an uncorrectable ECC error occurs	0x00

## FMC\_ECC\_INVALID\_ADDRL

FMC\_ECC\_INVALID\_ADDRL is the lower-4-byte ECC uncorrectable address register.

Offset Address: 0x00DC Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	fmc_ecc_invalid_addrl	Lower 4 bytes of the operation address of the flash that triggers the last interrupt when an uncorrectable ECC error occurs	0x0000_0000

## FMC\_BOOT\_CMD

FMC\_BOOT\_CMD is the boot read command configuration register.

Offset Address: 0x0037C Total Reset Value: 0x0000\_0003



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	rd_cmd	Boot read command 0x03: read (default) 0x0B: fast read. The frequency can be increased to 99 MHz at most. <b>dummy_num</b> needs to be configured synchronously. Other values: reserved	0x03



# Contents

<b>5 Network Interfaces .....</b>	<b>5-4</b>
5.1 ETH .....	5-4
5.1.1 Overview.....	5-4
5.1.2 Function Description .....	5-4
5.1.3 Operating Mode .....	5-5
5.1.4 Register Summary.....	5-12
5.1.5 Register Description .....	5-19
5.2 FE PHY .....	5-63
5.2.1 Overview.....	5-63
5.2.2 Features .....	5-63
5.2.3 Typical Application .....	5-64
5.2.4 Configuration.....	5-65
5.2.5 Summary of FEPHY_BASE Registers.....	5-69
5.2.6 Description of FEPHY_BASE Registers .....	5-70



## Figures

<b>Figure 5-1</b> Logic block diagram of the ETH module .....	5-5
<b>Figure 5-2</b> Process of receiving a frame in interrupt mode .....	5-7
<b>Figure 5-3</b> Process of receiving a frame in query mode .....	5-8
<b>Figure 5-4</b> Process of transmitting a frame by the CPU .....	5-9
<b>Figure 5-5</b> Typical application of the FE PHY.....	5-64



## Tables

<b>Table 5-1</b> Data structure of the frame descriptor received by the CPU.....	5-6
<b>Table 5-2</b> Data structure of the frame descriptor transmitted by the CPU .....	5-9
<b>Table 5-3</b> Summary of the MDIO control registers .....	5-12
<b>Table 5-4</b> Summary of the MAC controller registers .....	5-13
<b>Table 5-5</b> Summary of the global control registers.....	5-13
<b>Table 5-6</b> Summary of the statistics counter control registers.....	5-15
<b>Table 5-7</b> Summary of the statistics result registers .....	5-16
<b>Table 5-8</b> MDI pin description .....	5-64
<b>Table 5-9</b> Summary of the FEPHY_BASE registers .....	5-69



# 5 Network Interfaces

## 5.1 ETH

### 5.1.1 Overview

The Ethernet (ETH) module provides an ETH module interface that is used to receive data or transmit data through the network interface at a speed of 10 Mbit/s or 100 Mbit/s. This module also supports half-duplex or full-duplex operating mode and provides the media-independent interface (MII). With the eight configurable DMAC address filter tables, the ETH module filters input frames received through the network interface, limiting the traffic of the CPU port to protect the CPU against heavy traffic.

### 5.1.2 Function Description

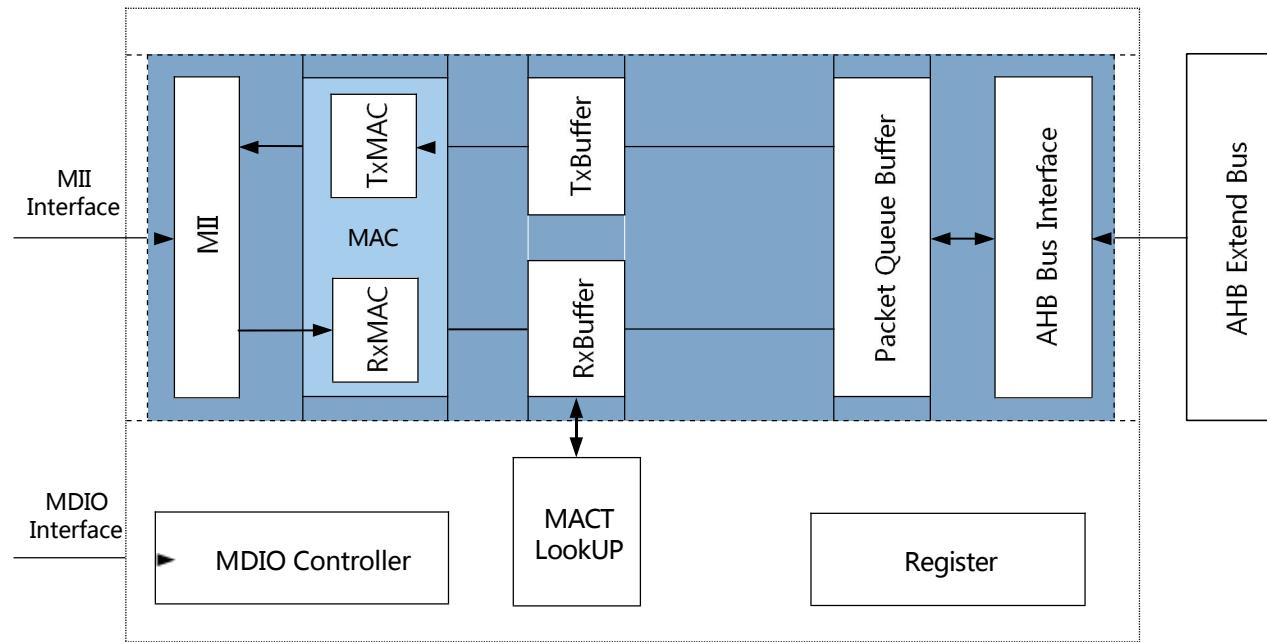
The ETH module has the following features:

- Supports one ETH module interface.
- Supports the rate of 10 Mbit/s or 100 Mbit/s.
- Supports full-duplex or half-duplex operating mode.
- Provides the MII for the internal PHY.
- Supports collision back-off and retransmission and late collision in half-duplex mode.
- Supports the transmission of flow control frames in full-duplex mode.
- Supports detection of frame length validity and the discarding of extra-long and extra-short frames.
- Implements cyclic redundancy check (CRC) on the input frames. The frames with CRC errors are discarded.
- Implements CRC check on the output frames
- Supports short-frame stuffing.
- Supports adaptation.

- Provides the management data input/output (MDIO) interfaces with configurable frequency.
- Provides 64 frame management queues for both data receive (RX) and data transmit (TX).
- Provides traffic limit to prevent the CPU against traffic attack.
- Supports the count of received frames and transmitted frames.
- Supports eight configurable DMAC address filter tables.
- Controls the transmission or discard of broadcast frames, multicast frames, and unicast frames.
- Supports the scatter gather (SG) function.
- Supports the checksum offload engine (COE) check and offload engine functions in the RX and TX directions.
- Supports Transmission Control Protocol (TCP) segmentation offload (TSO).
- Supports User Datagram Protocol (UDP) fragmentation offload (UFO).

Figure 5-1 shows the logic block diagram of the ETH module.

**Figure 5-1** Logic block diagram of the ETH module



## 5.1.3 Operating Mode

### 5.1.3.1 Process of Receiving Frames

During initialization, software needs to perform the following operations:



- Software needs to request a certain number of buffers. The number is equal to the RX queue depths and the size of each buffer is 2 KB. Then, software writes the header addresses of the buffers to the frame RX queue one by one. The times of the write operation is equal to the configured RX queue depth.
- The configured buffers should not be released during frame receiving. If the configured header address is not a word aligned address, the byte address corresponding to the header address must be a writable address.

The CPU performs the following steps when it is informed that a frame needs to be received:

- Step 1** Read the frame descriptor (including the start address and frame length of the RX frame) in the register [UD\\_GLB\\_IQFRM DES](#).
- Step 2** Process the data and write 1 to clear [GLB\\_IRQ\\_RAW](#)[iraw\_rx\_up] (indicating that the CPU completes the frame receiving).

----End

After receiving a frame of data, software needs to re-apply for a buffer of 2 KB and re-write the header address to the frame descriptor of the current RX queue. Otherwise, the available depth of RX queues equals to the number of buffers assigned to the CPU rather than the value configured by the CPU.

[Table 5-1](#)describes the data structure of the frame descriptor received by the CPU.

**Table 5-1** Data structure of the frame descriptor received by the CPU

Bits	Name	Description
[63:32]	rxfrm_saddr	Start address for receiving frames.
[31:18]	reserved	Reserved.
[17:12]	fd_in_addr	Relative address of the frame to be received in the input queue (IQ). It serves as the index (0 to iq_len - 1) of the absolute addresses for storing frames.
[11:0]	fd_in_len	Length of the frame to be received in the IQ.

### NOTE

The length of the RX queue can be obtained by querying [UD\\_GLB\\_ADDRQ\\_STAT](#).

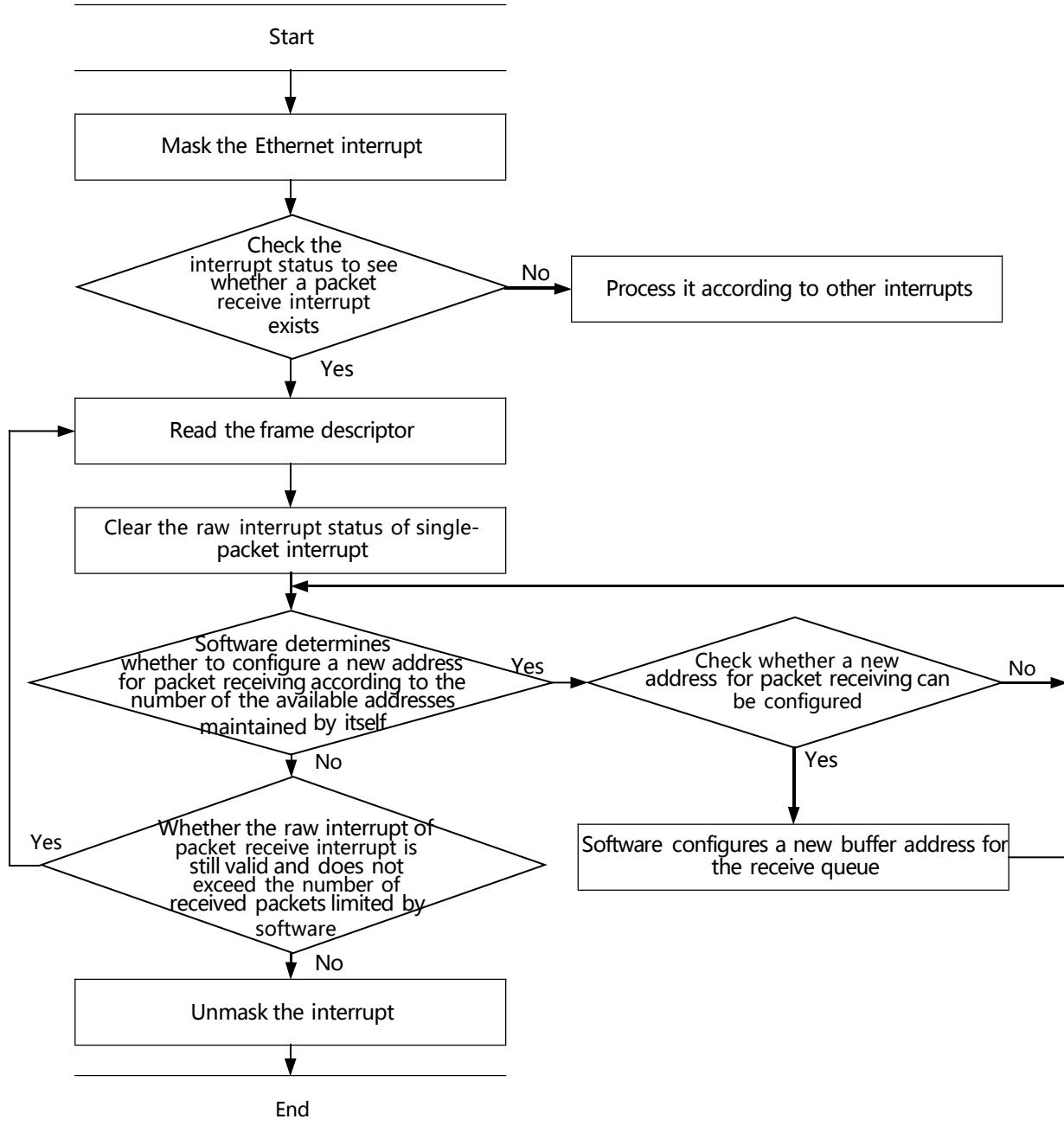
The CPU can receive a frame in interrupt or query mode.

#### 1. Receiving a frame in interrupt mode

When the CPU enables the frame RX interrupt, depending on the frames to be received, hardware generates frame RX interrupts (single-packet interrupt and multi-packet interrupt) int\_rx\_up and int\_rxd\_up.

int\_rx\_up indicates that an interrupt is reported each time a packet is received. int\_rxd\_up indicates that an interrupt is reported each time a number of specified packets are received. [Figure 5-2](#) shows the process of receiving a frame in interrupt mode.

**Figure 5-2** Process of receiving a frame in interrupt mode

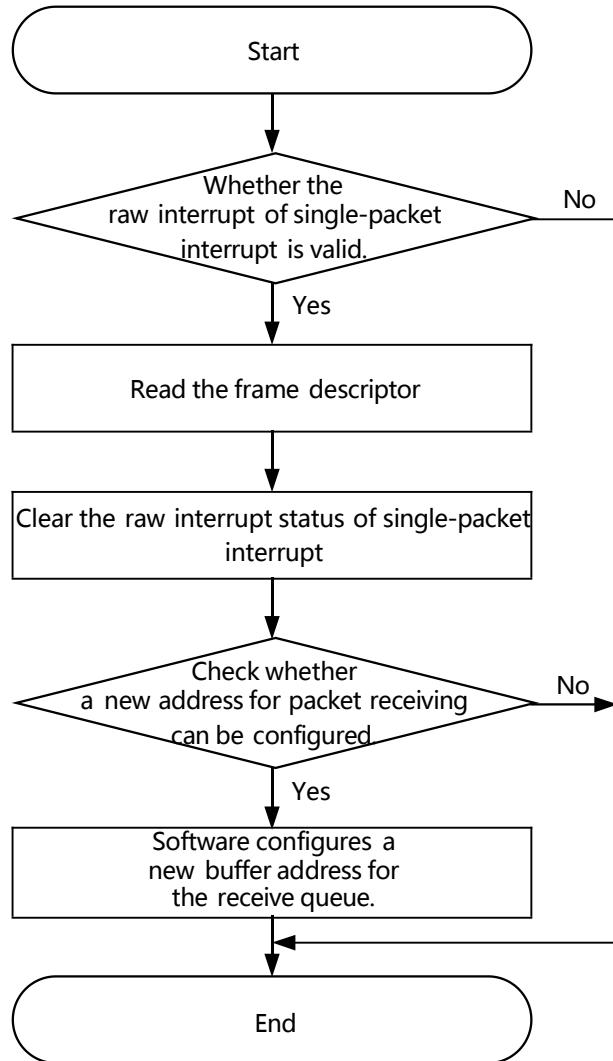


## 2. Receiving a frame in query mode

In this mode, the CPU does not enable the frame RX interrupt bit, namely, [GLB IRQ\\_ENA](#)[ien\_rx\_up], but automatically queries

[GLB\\_IRQ\\_RAW](#)[iraw\_rx\_up]. If [GLB\\_IRQ\\_RAW](#)[iraw\_rx\_up] is 1, it indicates that there is a frame to be received by the CPU. [Figure 5-3](#) shows the process of receiving a frame in query mode.

**Figure 5-3** Process of receiving a frame in query mode



### 5.1.3.2 Process of Transmitting a Frame

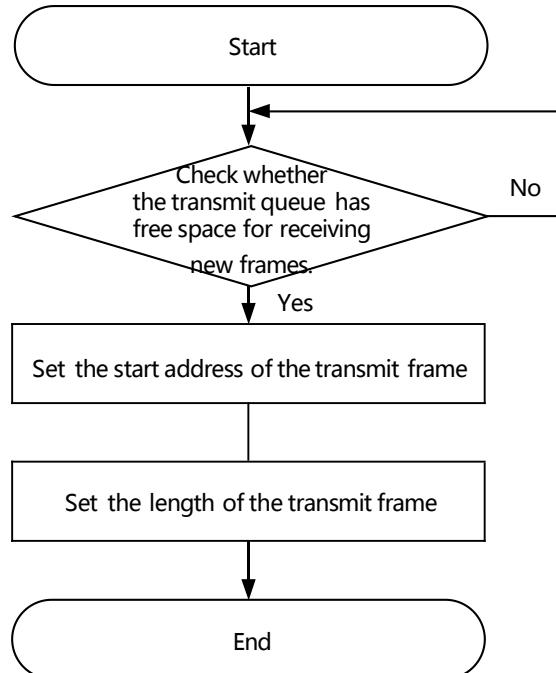
When a frame is transmitted, the CPU checks whether the current queue has any available space. If the space is sufficient, the CPU writes the header address of the buffer and then the length of the TX frame to the frame descriptor of the TX queue. The frame length trigger hardware for writing the TX frame writes the header address and frame length of the TX frame to the TX queue. Each time after a write is performed on the register, a data packet is transmitted. Therefore, software must control the write to the frame length register so that the frame length register is not written arbitrarily.

The frame format is as follows:

Destination MAC	Source MAC	Type	Data	FCS
-----------------	------------	------	------	-----

Figure 5-4 shows the process of transmitting a frame by the CPU.

**Figure 5-4** Process of transmitting a frame by the CPU



When the frame transmitted by the CPU is buffered in the SDRAM, the frame descriptor is not included. The frame descriptor is written to [UD\\_GLB\\_EQ\\_ADDR](#) and [UD\\_GLB\\_EQFRM\\_LEN](#) to notify the ETH module of adding the frame (descriptor) to the queue. [Table 5-2](#) describes the data structure of the frame descriptor transmitted by the CPU.

**Table 5-2** Data structure of the frame descriptor transmitted by the CPU

Bits	Name	Description
[63:32]	start_addr_eq	Header address of a frame to be transmitted
[31]	tso_flag	Whether to implement TSO 0: no 1: yes
[30]	vlan_flag	Whether the current packet has the virtual local area network (VLAN) flag 0: no 1: yes



Bits	Name	Description
[29]	ip_version	IP protocol version of the current packet 0: IPv4 1: IPv6
[28]	protocol_type	Transport layer protocol type of the current packet 0: TCP 1: UDP
[27]	tx_coe_flag	Whether to implement COE in the TX direction 0: no 1: yes
[26]	sg_flag	Whether to implement SG 0: no 1: yes
[25:24]	reserved	Reserved
[23:20]	ip_hdr_len	IP header length of the current packet, in the unit of word (four bytes) The IP header length is 5–15 words for an IPv4 packet and 10 words for an IPv6 packet.
[19:16]	protocol_hdr_len	TCP header length for a TCP packet, in the unit of word (four bytes), ranging from 5 words to 15 words; UDP header length for a UDP packet, in the unit of word (four bytes), 2 words
[15:11]	nfrags_num	Number of nfrags in the SKB of the current descriptor indicator. The maximum number is 16.
[10:0]	length	Maximum segment size (MSS) of the packet slice for the TSO packet, or total frame length for the SG or bypass packet

Note: The frames whose **length** is less than 20 bytes or greater than 1,900 bytes are discarded. In other words, the allowed range is from 20 bytes to 1900 bytes.

#### NOTE

The usage of the TX queues for the current CPU can be obtained by querying [UD\\_GLB\\_ADDRQ\\_STAT](#).

The CPU can transmit a frame in interrupt or query mode.



### 1. Transmitting a frame in interrupt mode

The CPU enables the nonempty-to-empty interrupt (int\_freeeq\_up) of the TX queue of the ETH module and allows the interrupt to be notified to the CPU. If the TX queue of the ETH module changes from nonempty to empty, it indicates the ETH module can transmit a frame. Then, hardware generates an interrupt to notify the CPU of transmitting the frame.

If software needs to transmit a frame but the current TX queue is full, software enables the interrupt. After the TX queue is empty, an interrupt is generated to instruct software to transmit the waiting frame. Software uses the interrupt to send a group of frames at a time and then releases the buffers for storing the previously transmitted group of frames when the interrupt is valid.

### 2. Transmitting a frame in query mode

Software queries the count of the TX frames. If the count is less than the configured depth of TX queue, it notifies the ETH module of the to-be transmitted frame directly. At the same time, it creates a corresponding TX frame index table whose content is the header address of the frame that is written to the TX queue of the Ethernet MAC. After transmitting a frame, the ETH module notifies the CPU of releasing the corresponding TX buffer through the address of the TX queue. Then, the CPU queries the corresponding TX buffer through the address of the TX queue and releases the buffer.

## 5.1.3.3 Interrupt Management

### Interrupt Status Register

This register indicates the generated interrupt type. For details, see [GLB\\_IRQ\\_STAT](#) in section [5.1.5.3 "Description of the Global Control Registers."](#)

### Interrupt Enable Register

This register controls the generation of related interrupts. For details, see [GLB\\_IRQ\\_ENA](#) in section [5.1.5.3 "Description of the Global Control Registers."](#) If an interrupt is enabled, the interrupt status is written to the related interrupt status register.

### Raw Interrupt Status Register

This register can read the raw interrupt of a type and transmit it to the CPU. For details, see [GLB\\_IRQ\\_RAW](#) in section [5.1.5.3 "Description of the Global Control Registers."](#) To clear the interrupt status, the raw interrupt of the interrupt must be cleared. After the raw interrupt is cleared, the interrupt status is cleared automatically.

### Traffic Control

When the number of frames received at a certain interval exceeds the upper limit configured by software, the subsequently received frames are selectively discarded.



Through the configuration of [UD\\_GLB\\_FC\\_DROPCTRL](#), the broadcast frames, multicast frames, or unicast frames are discarded if the traffic limit is exceeded. The traffic limit is configured through [UD\\_GLB\\_FC\\_RXLIMIT](#).

Software configures the time interval of traffic restriction through [UD\\_GLB\\_FC\\_TIMECTRL](#). For a 10-bit time interval register, the time slot can be set to up to 1,023. A 17-bit counter is used for counting the time slots of the main clock. The default count is 100,000. For a 100-MHz main clock, the time slot is 1 ms.

Software can configure the upper traffic limit of a 20-bit register. If the traffic limit is set to 0, it indicates that traffic is not limited.

## 5.1.4 Register Summary

### NOTE

Base address: 0x1029\_0000

## MDIO Control Registers

[Table 5-3](#)describes the MDIO control registers.

**Table 5-3** Summary of the MDIO control registers

Offset Address	Name	Description	Page
0x1100	MDIO_RWCTRL	MDIO_RWCTRL command word register	<a href="#">5-19</a>
0x1104	MDIO_RO_DATA	MDIO_RO_DATA read data register	<a href="#">5-20</a>
0x0108	UD_MDIO_PHYADDR	UD_MDIO_PHYADDR is a PHY physical address register.	<a href="#">5-21</a>
0x010C	UD_MDIO_RO_STAT	UD_MDIO_RO_STAT is a PHY status register.	<a href="#">5-21</a>
0x0114	UD_MDIO_IRQENA	UD_MDIO_IRQENA is a scan mask register for MDIO status changes.	<a href="#">5-22</a>

## MAC Control Registers

[Table 5-4](#)describes the MAC control registers.



**Table 5-4** Summary of the MAC controller registers

Offset Address	Name	Description	Page
0x0200	UD_MAC_PORTSEL	Port working status control register	<a href="#">5-23</a>
0x0204	UD_MAC_RO_STAT	Port status register	<a href="#">5-23</a>
0x0208	UD_MAC_PORTSET	Port working status configuration register	<a href="#">5-24</a>
0x020C	UD_MAC_STAT_CHANGE	Port status change indicator register for the MAC	<a href="#">5-24</a>
0x0210	UD_MAC_SET	MAC function configuration register	<a href="#">5-25</a>

## Global Control Registers

[Table 5-5](#)describes the Ethernet global control registers.

**Table 5-5** Summary of the global control registers

Offset Address	Name	Description	Page
0x1300	GLB_HOSTMAC_L32	Lower 32-bit register for the local MAC address	<a href="#">5-26</a>
0x1304	GLB_HOSTMAC_H16	Upper 16-bit register for the local MAC address	<a href="#">5-27</a>
0x1308	GLB_SOFT_RESET	Internal soft reset register	<a href="#">5-27</a>
0x1310	GLB_FWCTRL	Forward control register	<a href="#">5-27</a>
0x1314	GLB_MACTCTRL	MAC filter table control register	<a href="#">5-28</a>
0x1318	GLB_ENDIAN_MOD	Endian control register	<a href="#">5-29</a>
0x1330	GLB_IRQ_STAT	Interrupt status register	<a href="#">5-30</a>
0x1334	GLB_IRQ_ENA	Interrupt enable register	<a href="#">5-32</a>
0x1338	GLB_IRQ_RAW	Raw interrupt register	<a href="#">5-34</a>
0x1400	GLB_MAC0_L32	MAC filter 0	<a href="#">5-36</a>



Offset Address	Name	Description	Page
0x1404	GLB_MAC0_H16	MAC filter 0	<a href="#">5-36</a>
0x1408	GLB_MAC1_L32	MAC filter 1	<a href="#">5-36</a>
0x140C	GLB_MAC1_H16	MAC filter 1	<a href="#">5-37</a>
0x1410	GLB_MAC2_L32	MAC filter 2	<a href="#">5-37</a>
0x1414	GLB_MAC2_H16	MAC filter 2	<a href="#">5-37</a>
0x1418	GLB_MAC3_L32	MAC filter 3	<a href="#">5-38</a>
0x141C	GLB_MAC3_H16	MAC filter 3	<a href="#">5-38</a>
0x1420	GLB_MAC4_L32	MAC filter 4	<a href="#">5-39</a>
0x1424	GLB_MAC4_H16	MAC filter 4	<a href="#">5-39</a>
0x1428	GLB_MAC5_L32	MAC filter 5	<a href="#">5-40</a>
0x142C	GLB_MAC5_H16	MAC filter 5	<a href="#">5-40</a>
0x1430	GLB_MAC6_L32	MAC filter 6	<a href="#">5-41</a>
0x1434	GLB_MAC6_H16	MAC filter 6	<a href="#">5-41</a>
0x1438	GLB_MAC7_L32	MAC filter 7	<a href="#">5-41</a>
0x143C	GLB_MAC7_H16	MAC filter 7	<a href="#">5-42</a>
0x0340	UD_GLB_IRQN_SET	Multi-packet interrupt configuration register	<a href="#">5-42</a>
0x0344	UD_GLB_QLEN_SET	Queue length configuration register	<a href="#">5-43</a>
0x0348	UD_GLB_FC_LEVEL	Traffic control register	<a href="#">5-44</a>
0x034C	UD_GLB_CAUSE	Cause register for the CPU to which the packet is transmitted	<a href="#">5-44</a>
0x0350	UD_GLB_RXFRM_SADDR	RX frame start address register	<a href="#">5-45</a>
0x0354	UD_GLB_IQFRM_DES	RX frame descriptor register	<a href="#">5-45</a>
0x0358	UD_GLB_IQ_ADDR	RX frame header address register	<a href="#">5-46</a>



Offset Address	Name	Description	Page
0x035C	UD_GLB_BFC_STAT	Counter for traffic control status of forward buffer and aging time of multi-packet interrupt	<a href="#">5-46</a>
0x0360	UD_GLB_EQ_ADDR	TX queue header address register	<a href="#">5-47</a>
0x0364	UD_GLB_EQFRM_LEN	TX queue frame length configuration register	<a href="#">5-47</a>
0x0368	UD_GLB_QSTAT	Queue status register	<a href="#">5-48</a>
0x036C	UD_GLB_ADDRQ_STAT	Address queue status register	<a href="#">5-49</a>
0x0370	UD_GLB_FC_TIMECTRL	Traffic control time configuration register	<a href="#">5-50</a>
0x0374	UD_GLB_FC_RXLIMIT	Traffic control limit configuration register	<a href="#">5-50</a>
0x0378	UD_GLB_FC_DROPCTRL	Packet drop control register for traffic control	<a href="#">5-51</a>
0x0380	UD_GLB_RX_COE_EN	RX COE enable register	<a href="#">5-51</a>

## Statistics Counter Control Registers

[Table 5-6](#)describes the statistics counter control registers.

**Table 5-6** Summary of the statistics counter control registers

Offset Address	Name	Description	Page
0x0584	UD_STS_PORTCNT	Port status counter	<a href="#">5-52</a>
0x05A0	UD_PORT2CPU_PKTS	Register for the total number of packets received by the CPU from the uplink or downlink port	<a href="#">5-53</a>



Offset Address	Name	Description	Page
0x05A4	UD_CPU2IQ_ADDRCNT	Register for the count of configuring packet receiving address queue by the CPU	<a href="#">5-53</a>
0x05A8	UD_RX_IRQCNT	Register for the count of reporting single-packet interrupt by the uplink or downlink port	<a href="#">5-54</a>
0x05AC	UD_CPU2EQ_PKTS	Register for the total number of packets transmitted by the CPU to the uplink or downlink port	<a href="#">5-54</a>

## Statistics Result Registers

[Table 5-7](#) describes the statistics result registers.

**Table 5-7** Summary of the statistics result registers

Offset Address	Name	Description	Page
0x0600	UD_RX_DVCNT	RXDV rising edge count register	<a href="#">5-55</a>
0x0604	UD_RX_OCTS	Register for the total number of received bytes	<a href="#">5-55</a>
0x0608	UD_RX_RIGHTOCTS	Register for the total number of bytes of received correct packets	<a href="#">5-55</a>
0x060C	UD_HOSTMAC_PKTS	Register for the number of packets matching the local MAC address	<a href="#">5-56</a>
0x0610	UD_RX_RIGHTPKTS	Register for the total number of packets received by the port	<a href="#">5-56</a>
0x0614	UD_RX_BROADPKTS	Register for the number of correct broadcast packets	<a href="#">5-56</a>



Offset Address	Name	Description	Page
0x0618	UD_RX_MULTPKTS	Register for the number of correct multicast packets	<a href="#">5-57</a>
0x061C	UD_RX_UNIPKTS	Register for the number of correct unicast packets	<a href="#">5-57</a>
0x0620	UD_RX_ERRPKTS	Register for the total number of incorrect packets	<a href="#">5-57</a>
0x0624	UD_RX_CRCERR_PKTS	Register for the count of CRC errors	<a href="#">5-57</a>
0x0628	UD_RX_LENERR_PKTS	Register for the number of packets with invalid length	<a href="#">5-58</a>
0x062C	UD_RX_OCRCERR_PKTS	Register for the number of packets with odd nibbles and CRC errors	<a href="#">5-58</a>
0x0630	UD_RX_PAUSE_PKTS	Register for the number of received pause packets	<a href="#">5-58</a>
0x0634	UD_RF_OVERCNT	Register for the count of RXFIFO overflow events	<a href="#">5-59</a>
0x0638	UD_FLUX_TOL_IPKTS	Register for the total number of received packets allowed by the traffic limit	<a href="#">5-59</a>
0x063C	UD_FLUX_TOL_DPKTS	Register for the total number of packets discarded due to traffic limit	<a href="#">5-59</a>
0x064C	UD_MN2CPU_PKTS	Register for the number of packets not forwarded to the CPU port due to MAC limit	<a href="#">5-60</a>
0x0780	UD_TX_PKTS	Register for the total number of packets transmitted successfully	<a href="#">5-60</a>
0x0784	UD_TX_BROADPKTS	Register for the number of broadcast packets transmitted successfully	<a href="#">5-60</a>



Offset Address	Name	Description	Page
0x0788	UD_TX_MULTPKTS	Register for the number of multicast packets transmitted successfully	5-60
0x078C	UD_TX_UNIPKTS	Register for the number of unicast packets transmitted successfully	5-61
0x0790	UD_TX_OCTS	Register for the total number of transmitted bytes	5-61
0x0794	UD_TX_PAUSE_PKTS	Register for the number of transmitted pause frames	5-61
0x0798	UD_TX_RETRYCNT	Register for the total count of retransmission	5-62
0x079C	UD_TX_COLCNT	Register for the total count of collisions	5-62
0x07A0	UD_TX_LC_PKTS	Register for the number of packets with late collision	5-62
0x07A4	UD_TX_COLOK_PKTS	Register for the number of packets transmitted successfully with collisions	5-62
0x07A8	UD_TX_RETRY15_PKTS	Register for the number of packets discarded due to more than 15 times of retransmission	5-63
0x07AC	UD_TX_RETRYN_PKTS	Register for the number of packets with the count of collisions being equal to the threshold	5-63



## 5.1.5 Register Description

### 5.1.5.1 Description of the MDIO Control Registers

#### MDIO\_RWCTRL

MDIO\_RWCTRL is an MDIO command word register.

The register does not support soft reset.

Offset Address: 0x1100 Total Reset Value: 0x0000\_8000

Bits	Access	Name	Description	Reset
[31:16]	RW	cpu_data_in	Data used by the MDIO module to perform write operation on the PHY chip During write operation, the CPU first writes the 16-bit data to be written to the MDIO to this register.	0x0000
[15]	RW	finish	PHY read/write operation complete 0: not complete 1: complete When the read/write operation is required for the second time, the CPU must clear this bit first.	0x1
[14]	RO	reserved	Reserved	0x0
[13]	RW	rw	PHY read or write access control 0: read operation 1: write operation	0x0
[12:8]	RW	phy_exaddr	Physical address of the external PHY chip One MDIO can perform read/write operation on multiple external PHY chips. Each PHY chip has one corresponding address. When the MDIO connects to only one external PHY chip, this bit is equivalent to <a href="#">UD_MDIO_PHYADDR[phy0_addr]</a> or <a href="#">UD_MDIO_PHYADDR[phy1_addr]</a> .	0x00
[7:5]	RW	frq_dv	Frequency division factor for the MDC (the MDIO interface clock) when the MDIO performs the read/write operation on external PHY chips Take the frequency 100 MHz of the main	0x0



Bits	Access	Name	Description	Reset
			<p>clock as an example to describe the matching relations between frq_dv and MDC frequency.</p> <p>000: The frequency of the working main clock is divided by 50 and the obtained frequency is 2 MHz.</p> <p>001: The frequency of the working main clock is divided by 100 and the obtained frequency is 1 MHz.</p> <p>010: The frequency of the working main clock is divided by 200 and the obtained frequency is 500 kHz.</p> <p>011: The frequency of the working main clock is divided by 400 and the obtained frequency is 250 kHz.</p> <p>100: The frequency of the working main clock is divided by 800 and the obtained frequency is 125 kHz.</p> <p>101: The frequency of the working main clock is divided by 1600 and the obtained frequency is 62.5 kHz.</p> <p>110: The frequency of the working main clock is divided by 3200 and the obtained frequency is 31.25 kHz.</p> <p>111: The frequency of the working main clock is divided by 6400 and the obtained frequency is 15.625 kHz.</p>	
[4:0]	RW	phy_inaddr	Internal register address of the external PHY chip. This address is presented by a 5-bit binary number.	0x00

## MDIO\_RO\_DATA

MDIO\_RO\_DATA is MDIO\_RO\_DATAMDIO is a read data register. The register does not support soft reset.

Offset Address: 0x1104 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	Reserved	0x0000



Bits	Access	Name	Description	Reset
[15:0]	RO	cpu_data_out	Data register used by the MDIO module to perform read operation on the PHY chip. The MDIO module first writes the 16-bit data read from the PHY chip to this register.	0x0000

## UD\_MDIO\_PHYADDR

UD\_MDIO\_PHYADDR is a PHY physical address register. The register does not support soft reset.

Offset Address: 0x0108 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:5]	RO	reserved	Reserved	0x00000000
[4:0]	RW	phy_addr	Physical address of the external PHY chip	0x01

## UD\_MDIO\_RO\_STAT

UD\_MDIO\_RO\_STAT is a PHY status register. The register does not support soft reset.

Offset Address: 0x010C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	Reserved	0x00000000
[2]	RO	speed_mdio2mac	Port speed working status obtained from the MDIO interface, which is in either 10 Mbit/s or 100 Mbit/s working mode 0: 10 Mbit/s mode 1: 100 Mbit/s mode	0x0
[1]	RO	link_mdio2mac	Port link status obtained from the MDIO interface 0: No link exists. 1: A link exists.	0x0
[0]	RO	duplex_mdio2mac	Port duplex working status obtained from the MDIO interface 0: half-duplex	0x0



Bits	Access	Name	Description	Reset
			1: full-duplex	

## UD\_MDIO\_IRQENA

UD\_MDIO\_IRQENA is a scan mask register for MDIO status changes. The register does not support soft reset.

### NOTE

link\_partner status change refers to the change of any bit of link, speed, and duplex for the PHY status.

Offset Address: 0x0114 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	RO	reserved	Reserved	0x00000000
[3]	RW	link_partner_ch_mask	Port link partner status scan change interrupt mask 0: mask 1: unmask	0x0
[2]	RW	speed_ch_mask	Port speed mode scan change interrupt mask 0: mask 1: unmask	0x0
[1]	RW	link_ch_mask	Port link mode scan change interrupt mask 0: mask 1: unmask	0x0
[0]	RW	duplex_ch_mask	Port duplex mode scan change interrupt mask 0: mask 1: unmask	0x0

### 5.1.5.2 Description of the MAC Control Registers

MAC control registers are registers for port control. When the port status is valid, after configuring MAC control registers, you need to perform one soft reset on them.



## UD\_MAC\_PORTSEL

UD\_MAC\_PORTSEL is a port working status control register. The register does not support soft reset.

Offset Address: 0x0200 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:2]	RO	reserved	Reserved	0x00000000
[1]	RW	mii	Port interface mode select 0: MII interface 1: reserved	0x0
[0]	RW	stat_ctrl	Port working status information select control register 0: Use the status information obtained from the MDIO interface. 1: Use the status information set by the CPU.	0x1

## UD\_MAC\_RO\_STAT

UD\_MAC\_RO\_STAT is a port status register. The register does not support soft reset.

Offset Address: 0x0204 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	Reserved	0x00000000
[2]	RO	speed_stat	Port current speed mode 0: 10 Mbit/s mode 1: 100 Mbit/s mode	0x0
[1]	RO	link_stat	Port current link status 0: No link exists. 1: A link exists.	0x0
[0]	RO	duplex_stat	Port current duplex status 0: half-duplex 1: full-duplex	0x0



## UD\_MAC\_PORTSET

UD\_MAC\_PORTSET is a port working status configuration register. The register does not support soft reset.

Offset Address: 0x0208 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	Reserved	0x00000000
[2]	RW	speed_stat_dio	Port speed mode set by the CPU 0: 10 Mbit/s mode 1: 100 Mbit/s mode	0x0
[1]	RW	link_stat_dio	Port link status set by the CPU 0: No link exists. 1: A link exists.	0x0
[0]	RW	duplex_stat_dio	Port duplex mode set by the CPU 0: half-duplex 1: full-duplex	0x0

## UD\_MAC\_STAT\_CHANGE

UD\_MAC\_STAT\_CHANGE is a port status change indicator register. The register does not support soft reset.

Offset Address: 0x020C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	Reserved	0x00000000
[2]	WC	speed_stat_ch	Port speed mode change indicator 0: No change occurs. 1: A change occurs. Writing 1 clears this register.	0x0
[1]	WC	link_stat_ch	Port link status change indicator 0: No change occurs. 1: A change occurs. Writing 1 clears this register.	0x0
[0]	WC	duplex_stat_ch	Port duplex mode change indicator 0: No change occurs.	0x0



Bits	Access	Name	Description	Reset
			1: A change occurs. Writing 1 clears this register.	

## UD\_MAC\_SET

UD\_MAC\_SET is a MAC function configuration register.

The register does not support soft reset.

Offset Address: 0x0210 Total Reset Value: 0x2027\_55EE

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	Reserved	0x0
[29]	RW	add_pad_en	Port auto add PAD enable during transmission 0: disabled 1: enabled	0x1
[28]	RW	crcgen_dis	Port CRC generation disable control 0: CRC is recalculated for the output frame. 1: CRC is not recalculated for the output frame.	0x0
[27]	RW	cntr_rdclr_en	Port statistics counter read clear enable 0: disabled 1: enabled	0x0
[26]	RW	cntr_clr_all	Port statistics counter clear control 0: not clear 1: clear <b>NOTE</b> If cntr_clr_all is set to 1, the next clear all operation can be performed only after this bit is set to 0 and then to 1.	0x0
[25]	RW	cntr_roll_dis	Port statistics acyclic counter enable 0: disabled 1: enabled	0x0
[24:21]	RW	colthreshold	Port collision count statistics threshold The default value is 0x1, which indicates	0x1



Bits	Access	Name	Description	Reset
			the count of frames with one collision.	
[20]	-	reserved	Reserved	0x0
[19]	-	reserved	Reserved	0x0
[18]	RW	pause_en	Port pause frame TX enable 0: disabled 1: enabled	0x1
[17]	RW	rx_shframe_en	Port short frame RX enable 0: disabled 1: enabled <b>NOTE</b> If rx_shframe_en is set to 1, the minimum RX frame length allowed by the port is that set by rx_min_thr. If rx_shframe_en is set to 0, the minimum RX frame length allowed by the port is 64 bytes (including CRC) by default.	0x1
[16:11]	RW	rx_min_thr	Minimum RX frame length allowed by the port The value range is from 42 bytes to 63 bytes. The default value is 42 bytes. <b>NOTE</b> If rx_min_thr is set to a value smaller than 42, 42 is used instead of the value.	0x2A
[10:0]	RW	len_max	Maximum RX frame length allowed by the port. The default value is 1518 bytes. The value is in a range of 1518 bytes to 1535 bytes. <b>NOTE</b> If len_max is set to a value greater than 2000, 2000 is used instead of the value. If len_max is set to a value smaller than 256, 256 is used instead of the value.	0x5EE

### 5.1.5.3 Description of the Global Control Registers

#### GLB\_HOSTMAC\_L32

GLB\_HOSTMAC\_L32 is a lower 32-bit register for the local MAC address.



The register does not support soft reset.

Offset Address: 0x1300 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	local_mac	Lower 32 bits of the local MAC address	0x00000000

## GLB\_HOSTMAC\_H16

GLB\_HOSTMAC\_H16 is an upper 16-bit register for the local MAC address.

The register does not support soft reset.

Offset Address: 0x1304 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	Reserved	0x0000
[15:0]	RW	local_mac[47:32]	Upper 16 bits of the local MAC address	0x0000

## GLB\_SOFT\_RESET

GLB\_SOFT\_RESET is an internal soft reset register.

The register does not support soft reset.

### NOTE

The time for each soft reset must remain for more than 2 ms.

Offset Address: 0x1308 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	Reserved	0x00000000
[0]	RW	soft_reset	Internal soft reset 0: not reset 1: reset In soft reset state, this bit must be set to 0 to clear soft reset.	0x0

## GLB\_FWCTRL

GLB\_FWCTRL is a forward control register.



The register does not support soft reset.

Offset Address: 0x1310 Total Reset Value: 0x0000\_0020

Bits	Access	Name	Description	Reset
[31:8]	RO	reserved	Reserved	0x000000
[7]	RW	fwall2cpu_up	Whether to forcibly forward all valid input frames to the CPU port 0: no 1: yes	0x0
[6]	RO	reserved	Reserved	0x0
[5]	RW	fw2cpu_ena_up	Function enable of forwarding the input frames to the CPU port 0: disabled 1: enabled	0x1
[4:0]	RO	reserved	Reserved	0x00

## GLB\_MACTCTRL

GLB\_MACTCTRL is a MAC filter table control register.

The register does not support soft reset.

### NOTE

- If the highest byte of the destination MAC address is even, the frame is a unicast frame.
- If the highest byte of the destination MAC address is odd, the frame is a multicast frame.
- If all bytes of the destination MAC address are 0xFF, the frame is a broadcast frame.

Offset Address: 0x1314 Total Reset Value: 0x0000\_0020

Bits	Access	Name	Description	Reset
[31:8]	RO	reserved	Reserved	0x000000
[7]	RW	mact_ena_up	Enable bit of all MAC filters of the port 0: disabled (no MAC filter is used) 1: enabled (MAC filters are used)	0x0
[6]	RO	reserved	Reserved	0x0
[5]	RW	broad2cpu_up	Whether to forward the input broadcast frames to the CPU port 0: no	0x1



Bits	Access	Name	Description	Reset
			1: yes	
[4]	RO	reserved	Reserved	0x0
[3]	RW	multi2cpu_up	Whether to forward the input multicast frames that are not listed in the filter table to the CPU port  0: no 1: yes	0x0
[2]	RO	reserved	Reserved	0x0
[1]	RW	uni2cpu_up	Whether to forward the input unicast frames that are not listed in the filter table to the CPU port  0: no 1: yes	0x0
[0]	RO	reserved	Reserved	0x0

## GLB\_ENDIAN\_MOD

GLB\_ENDIAN\_MOD is an endian control register.

The register does not support soft reset.

Offset Address: 0x1318 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:2]	RO	reserved	Reserved	0x00000000
[1]	RW	in_endian	RX packet write SDRAM endian configuration  0: big-endian mode 1: little-endian mode Data consists of bytes	0x1
[0]	RW	out_endian	TX packet read SDRAM endian configuration  0: big-endian mode 1: little-endian mode	0x1



## GLB\_IRQ\_STAT

GLB\_IRQ\_STAT is an interrupt status register.

The register does not support soft reset.

Offset Address: 0x1330 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:13]	RO	reserved	Reserved	0x00000
[12]	RO	int_mdio_finish	Raw interrupt status indicating whether the MDIO interface completes the operation required by the CPU 0: not completed 1: Completed and an interrupt is generated. After this interrupt is generated, software determines whether the MDIO completes the operation by querying <a href="#">MDIO_RWCTRL[finish]</a> .	0x0
[11:8]	RO	reserved	Reserved	0x0
[7]	RO	int_rxd_up	Interrupt status (multi-packet interrupt) for frames on the port to be received by the CPU 0: The interrupt is invalid. 1: The interrupt is valid. There are frames to be received by the CPU in the RX queue. After this interrupt is generated, software determines whether there are frames to be received by querying <a href="#">GLB_IRQ_RAW[iraw_rxd_up]</a> .	0x0
[6]	RO	int_freeeq_up	Interrupt status indicates that the status of the port output queue is changed from nonempty to empty, that is, the status of the TX queue buffer is changed from nonempty to empty, that is, the status of the TX queue buffer is changed from nonempty to empty. In this case, the CPU can write a group of new frames to be transmitted. 0: No interrupt is generated. 1: An interrupt is generated. After this interrupt is generated, software	0x0



Bits	Access	Name	Description	Reset
			determines whether the current TX queue is empty by querying <a href="#">UD_GLB_ADDRQ_STAT[eq_cnt]</a> . If the current TX queue is not empty, it indicates that the interrupt is invalid.	
[5]	RO	int_stat_up	Interrupt status for port status changes, indicating that an interrupt is generated when the MDIO obtains the speed change, duplex mode change, and link status change of the PHY chip in auto-adaption mode. 0: The interrupt is invalid. 1: The interrupt is valid. The port status changes. After this interrupt is generated, software determines which status changes according to the configuration of <a href="#">UD_MDIO_IRQENA</a> .	0x0
[4]	RO	int_duplex_up	Interrupt status for port duplex mode changes 0: The interrupt is invalid. 1: The interrupt is valid. The duplex mode changes. After this interrupt is generated, software determines whether the duplex mode changes by querying <a href="#">UD_MAC_STAT_CHANGE[duplex_stat_ch]</a> .	0x0
[3]	RO	int_speed_up	Interrupt status for port speed mode changes 0: The interrupt is invalid. 1: The interrupt is valid. The speed mode changes. After this interrupt is generated, software determines whether the speed mode changes by querying <a href="#">UD_MAC_STAT_CHANGE[speed_stat_ch]</a> .	0x0
[2]	RO	int_link_up	Interrupt status for port link status changes 0: The interrupt is invalid. 1: The interrupt is valid. The link status	0x0



Bits	Access	Name	Description	Reset
			changes. After this interrupt is generated, software determines whether the link status changes by querying <a href="#">UD_MAC_STAT_CHANGE</a> [link_stat_ch].	
[1]	RO	int_tx_up	Interrupt status for the completion of transmitting a frame from the CPU by the port 0: not completed 1: Completed and an interrupt is generated. After this interrupt is generated, software determines whether to release the buffer of the TX frames by querying the current TX queue address eq_out_index in <a href="#">UD_GLB_QSTAT</a> .	0x0
[0]	RO	int_rx_up	Interrupt status for frames on the port to be received by the CPU 0: The interrupt is invalid. 1: The interrupt is valid. There are frames to be received by the CPU in the RX queue. After this interrupt program is started, software determines whether frames are received by querying the <a href="#">GLB_IRQ_RAW</a> [iraw_rxd_up] signal.	0x0

## GLB\_IRQ\_ENA

GLB\_IRQ\_ENA is an interrupt enable register.

The register does not support soft reset.

Offset Address: 0x1334 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:20]	RO	reserved	Reserved	0x000
[19]	RW	ien_all	All interrupts enable 0: disabled (none of the interrupt can be reported) 1: enabled (all interrupts are reported)	0x0



Bits	Access	Name	Description	Reset
			according to the configuration)	
[18]	RW	ien_up	All uplink port interrupts enable 0: disabled (none of the uplink port interrupt can be reported) 1: enabled (all uplink port interrupts are reported according to the configuration)	0x0
[17:13]	RO	reserved	Reserved	0x00
[12]	RW	ien_mdio_finish	Indicator enable for the MDIO to complete the operation required by the CPU 0: disabled 1: enabled	0x0
[11:8]	RO	reserved	Reserved	0x0
[7]	RW	ien_rxd_up	Interrupt enable (multi-packet interrupt) for a frame (frames) on the uplink port to be received by the CPU 0: disabled 1: enabled	0x0
[6]	RW	ien_freeeq_up	Interrupt signal enable for the TX queue of the uplink port to change from nonempty to empty 0: disabled 1: enabled	0x0
[5]	RW	ien_stat_up	Interrupt signal enable for uplink port status changes 0: disabled 1: enabled	0x0
[4]	RW	ien_duplex_up	Interrupt enable for uplink port duplex mode changes 0: disabled 1: enabled	0x0
[3]	RW	ien_speed_up	Interrupt enable for uplink port speed mode changes 0: disabled 1: enabled	0x0
[2]	RW	ien_link_up	Interrupt enable for uplink port link	0x0



Bits	Access	Name	Description	Reset
			status changes 0: disabled 1: enabled	
[1]	RW	ien_tx_up	Indicator enable for the completion of transmitting a frame from the CPU by the uplink port 0: disabled 1: enabled	0x0
[0]	RW	ien_rx_up	Interrupt enable for frames on the uplink port to be received by the CPU 0: disabled 1: enabled	0x0

## GLB\_IRQ\_RAW

GLB\_IRQ\_RAW is a raw interrupt register. The register does not support soft reset. Writing 1 clears this register.

Offset Address: 0x1338 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:13]	RO	reserved	Reserved	0x00000
[12]	WC	iraw_mdio_finish	Raw interrupt status for the MDIO to complete the operation required by the CPU 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[11:8]	RO	reserved	Reserved	0x0
[7]	WC	iraw_rxd_up	Raw interrupt status (multi-packet interrupt) for a frame (frames) on the uplink port to be received by the CPU 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	WC	iraw_freeeq_up	Raw interrupt status for the TX queue of the uplink port to change from nonempty to empty, indicating that the TX queue buffer changes from nonempty to empty and the CPU can write a group of new	0x0



Bits	Access	Name	Description	Reset
			frames to be transmitted. 0: No interrupt is generated. 1: An interrupt is generated.	
[5]	WC	iraw_stat_up	Raw interrupt status for uplink port status changes, indicating that an interrupt is generated when the MDIO obtains the speed change, duplex mode change, and link status change of the PHY chip in auto-adaption mode. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	WC	iraw_duplex_up	Raw interrupt status for uplink port duplex mode changes 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	WC	iraw_speed_up	Raw interrupt status for uplink port speed mode changes 0: The interrupt is invalid. 1: The interrupt is valid. The speed mode changes. Writing 1 clears this register.	0x0
[2]	WC	iraw_link_up	Raw interrupt status for uplink port link status changes 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	WC	iraw_tx_up	Raw interrupt status for the completion of transmitting a frame from the CPU by the uplink port 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	WC	iraw_rx_up	Raw interrupt status for frames on the uplink port to be received by the CPU 0: No interrupt is generated. 1: An interrupt is generated.	0x0



## GLB\_MAC0\_L32

GLB\_MAC0\_L32 is a lower 32-bit register for the filter table MAC0.

Offset Address: 0x1400 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac0	Lower 32 bits of the filter table MAC0	0x00000000

## GLB\_MAC0\_H16

GLB\_MAC0\_H16 is an upper 16-bit register for the filter table MAC0.

Offset Address: 0x1404 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	Reserved	0x000
[21]	RW	fw2cpu_up	Whether to forward the frames received by the uplink port that match this filter to the CPU port when the uplink port enables this filter  0: no 1: yes	0x0
[20:18]	RO	reserved	Reserved	0x0
[17]	RW	mac0_up	Control for setting this filter to be used by the uplink port  0: The uplink port does not use this filter. 1: The uplink port uses this filter.	0x0
[16]	RO	reserved	Reserved	0x0
[15:0]	RW	flt_mac0	Upper 16 bits of the filter table MAC0	0x0000

## GLB\_MAC1\_L32

GLB\_MAC1\_L32 is a lower 32-bit register for the filter table MAC1.

Offset Address: 0x1408 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac1	Lower 32 bits of the filter table MAC1	0x00000000



## GLB\_MAC1\_H16

GLB\_MAC1\_H16 is an upper 16-bit register for the filter table MAC1.

Offset Address: 0x140C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	Reserved	0x000
[21]	RW	fw2cpu_up	Whether to forward the frames received by the uplink port that match this filter to the CPU port when the uplink port enables this filter  0: no 1: yes	0x0
[20:18]	RO	reserved	Reserved	0x0
[17]	RW	mac1_up	Control for setting this filter to be used by the uplink port  0: The uplink port does not use this filter. 1: The uplink port uses this filter.	0x0
[16]	RO	reserved	Reserved	0x0
[15:0]	RW	flt_mac1	Upper 16 bits of the filter table MAC1	0x0000

## GLB\_MAC2\_L32

GLB\_MAC2\_L32 is a lower 32-bit register for the filter table MAC2.

Offset Address: 0x1410 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac2	Lower 32 bits of the filter table MAC2	0x00000000

## GLB\_MAC2\_H16

GLB\_MAC2\_H16 is an upper 16-bit register for the filter table MAC2.

Offset Address: 0x1414 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	Reserved	0x000
[21]	RW	fw2cpu_up	Whether to forward the frames received by the uplink port that match this filter to the CPU port when the uplink port enables this filter 0: no 1: yes	0x0
[20:18]	RO	reserved	Reserved	0x0
[17]	RW	mac2_up	Control for setting this filter to be used by the uplink port 0: The uplink port does not use this filter. 1: The uplink port uses this filter.	0x0
[16]	RO	reserved	Reserved	0x0
[15:0]	RW	flt_mac2	Upper 16 bits of the filter table MAC2	0x0000

## GLB\_MAC3\_L32

GLB\_MAC3\_L32 is a lower 32-bit register for the filter table MAC3.

Offset Address: 0x1418 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac3	Lower 32 bits of the filter table MAC3	0x00000000

## GLB\_MAC3\_H16

GLB\_MAC3\_H16 is an upper 16-bit register for the filter table MAC3.

Offset Address: 0x141C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	Reserved	0x000
[21]	RW	fw2cpu_up	Whether to forward the frames received by the uplink port that match this filter to the CPU port when the uplink port enables this filter 0: no	0x0



Bits	Access	Name	Description	Reset
			1: yes	
[20:18]	RO	reserved	Reserved	0x0
[17]	RW	mac3_up	Whether to forward the frames received by the downlink port that match this filter to the CPU port when the downlink port enables this filter  0: Do not forward the frames. 1: Forward the frames.	0x0
[16]	RO	reserved	Reserved	0x0
[15:0]	RW	flt_mac3	Upper 16 bits of the filter table MAC3	0x0000

## GLB\_MAC4\_L32

GLB\_MAC4\_L32 is a lower 32-bit register for the filter table MAC4.

Offset Address: 0x1420 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac4	Lower 32 bits of the filter table MAC4	0x00000000

## GLB\_MAC4\_H16

GLB\_MAC4\_H16 is an upper 16-bit register for the filter table MAC4.

Offset Address: 0x1424 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	Reserved	0x000
[21]	RW	fw2cpu_up	Whether to forward the frames received by the uplink port that match this filter to the CPU port when the uplink port enables this filter  0: no 1: yes	0x0
[20:18]	RO	reserved	Reserved	0x0
[17]	RW	mac4_up	Control for setting this filter to be used by the uplink port	0x0



Bits	Access	Name	Description	Reset
			0: The uplink port does not use this filter. 1: The uplink port uses this filter.	
[16]	RO	reserved	Reserved	0x0
[15:0]	RW	flt_mac4	Upper 16 bits of the filter table MAC4	0x0000

## GLB\_MAC5\_L32

GLB\_MAC5\_L32 is a lower 32-bit register for the filter table MAC5.

Offset Address: 0x1428 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac5	Lower 32 bits of the filter table MAC5	0x00000000

## GLB\_MAC5\_H16

GLB\_MAC5\_H16 is an upper 16-bit register for the filter table MAC5.

Offset Address: 0x142C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	Reserved	0x000
[21]	RW	fw2cpu_up	Whether to forward the frames received by the uplink port that match this filter to the CPU port when the uplink port enables this filter  0: no 1: yes	0x0
[20:18]	RO	reserved	Reserved	0x0
[17]	RW	mac5_up	Control for setting this filter to be used by the uplink port  0: The uplink port does not use this filter. 1: The uplink port uses this filter.	0x0
[16]	RO	reserved	Reserved	0x0
[15:0]	RW	flt_mac5	Upper 16 bits of the filter table MAC5	0x0000



## GLB\_MAC6\_L32

GLB\_MAC6\_L32 is a lower 32-bit register for the filter table MAC6.

Offset Address: 0x1430 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac6	Lower 32 bits of the filter table MAC6	0x00000000

## GLB\_MAC6\_H16

GLB\_MAC6\_H16 is an upper 16-bit register for the filter table MAC6.

Offset Address: 0x1434 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	Reserved	0x000
[21]	RW	fw2cpu_up	Whether to forward the frames received by the uplink port that match this filter to the CPU port when the uplink port enables this filter  0: no 1: yes	0x0
[20:18]	RO	reserved	Reserved	0x0
[17]	RW	mac6_up	Control for setting this filter to be used by the uplink port  0: The uplink port does not use this filter. 1: The uplink port uses this filter.	0x0
[16]	RO	reserved	Reserved	0x0
[15:0]	RW	flt_mac6	Upper 16 bits of the filter table MAC6	0x0000

## GLB\_MAC7\_L32

GLB\_MAC7\_L32 is a lower 32-bit register for the filter table MAC7.

Offset Address: 0x1438 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	flt_mac7	Lower 32 bits of the filter table MAC7	0x00000000



## GLB\_MAC7\_H16

GLB\_MAC7\_H16 is an upper 16-bit register for the filter table MAC7.

Offset Address: 0x143C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	RO	reserved	Reserved	0x000
[21]	RW	fw2cpu_up	Whether to forward the frames received by the uplink port that match this filter to the CPU port when the uplink port enables this filter  0: no 1: yes	0x0
[20:18]	RO	reserved	Reserved	0x0
[17]	RW	mac7_up	Control for setting this filter to be used by the uplink port  0: The uplink port does not use this filter. 1: The uplink port uses this filter.	0x0
[16]	RO	reserved	Reserved	0x0
[15:0]	RW	flt_mac7	Upper 16 bits of the filter table MAC7	0x0000

## UD\_GLB\_IRQN\_SET

UD\_GLB\_IRQN\_SET is a multi-packet interrupt configuration register.

The register does not support soft reset.

Offset Address: 0x0340 Total Reset Value: 0x0800\_003A

Bits	Access	Name	Description	Reset
[31:29]	RO	reserved	Reserved	0x0
[28:24]	RW	int_frm_cnt	These bits are used to set the multi-packet interrupt function. That is, how many packets must be received before a multi-packet interrupt can be reported.  <b>NOTE</b>  The minimum value of int_frm_cnt can be set	0x08



Bits	Access	Name	Description	Reset
			to 1. In this case, multi-packet interrupt is equivalent to single-packet interrupt.	
[23:16]	RO	reserved	Reserved	0x00
[15:0]	RW	age_timer	After the multi-packet interrupt function is enabled, if the number of received packets cannot reach the specified number of packets required for reporting the multi-packet interrupt after a period, this period is defined as the aging time for generating the multi-packet interrupt.  <b>NOTE</b> age_timer is counted in the unit of the main clock cycle divided by 256.	0x003A

## UD\_GLB\_QLEN\_SET

UD\_GLB\_QLEN\_SET is a queue length configuration register.

The register does not support soft reset.

Offset Address: 0x0344 Total Reset Value: 0x0000\_2020

Bits	Access	Name	Description	Reset
[31:14]	RO	reserved	Reserved	0x00000
[13:8]	RW	iq_len	RX (packet RX) queue length configuration  <b>NOTE</b> iq_len cannot be set to 0. Otherwise, it is forcibly set to 1. The sum of the set values of iq_len and eq_len cannot be greater than 64. Otherwise, the value (non-zero) of iq_len is first assigned and the value of eq_len is calculated by the formula: 64 - iq_len.	0x20
[7:6]	RO	reserved	Reserved	0x0
[5:0]	RW	eq_len	TX (packet TX) queue length configuration  <b>NOTE</b> eq_len cannot be set to 0. Otherwise, it is forcibly set to 1.	0x20



## UD\_GLB\_FC\_LEVEL

UD\_GLB\_FC\_LEVEL is a traffic control register.

The register does not support soft reset.

Offset Address: 0x0348 Total Reset Value: 0x3018\_0508

Bits	Access	Name	Description	Reset
[31:15]	RO	reserved	Reserved	0x06030
[14]	RW	qlimit_ena	Traffic control enable for RX queue 0: Disabled (do not transmit the traffic control message according to the status of RX queue). 1: Enabled (transmit the traffic control message according to the status of RX queue).	0x0
[13:8]	RW	qlimit_up	Upper limit of traffic control for RX queue. When the free space of the RX queue is less than the upper limit, if traffic control for RX queue is enabled, the traffic control message is transmitted to the peer end.   <b>NOTE</b> <ul style="list-style-type: none"><li>⌚ If the upper limit qlimit_up is set to 0, the RX queue fails to enter the traffic control status.</li><li>⌚ The upper limit qlimit_up must be greater than the lower limit qlimit_down.</li></ul>	0x05
[7:6]	RO	reserved	Reserved	0x0
[5:0]	RW	qlimit_down	Lower limit of traffic control for RX queue. When the free space of the RX queue is greater than or equal to the upper limit, if the RX queue is in traffic control state, the current traffic control is stopped.	0x08

## UD\_GLB\_CAUSE

UD\_GLB\_CAUSE is a cause register for the CPU to which the packet is transmitted.

The register does not support soft reset.



Offset Address: 0x034C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	Reserved	0x00000000
[2:0]	RO	mact_cause	Packet matching result types by querying the MAC table 000: forced forwarding 001: Packet whose destination MAC address is the local MAC address. 010: broadcast packet 011: packet matching the MAC table 100: multicast packet not matching the MAC table 101: unicast packet not matching the MAC table Others: reserved	0x0

## UD\_GLB\_RXFRM\_SADDR

UD\_GLB\_RXFRM\_SADDR is an RX frame start address register.

The register does not support soft reset.

Bits	Access	Name	Description	Reset
[31:0]	RO	rxfrm_saddr	Start address of the RX frame	0x00000000

## UD\_GLB\_IQFRM\_DES

UD\_GLB\_IQFRM\_DES is an RX frame descriptor register.

The register does not support soft reset.

Offset Address: 0x0354 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:18]	RO	reserved	Reserved	0x0000
[17:12]	RO	fd_in_addr	Relative address of the first frame to be received in the input queue (IQ). It serves as the index (0 to iq_len-1) of the absolute address for storing the frames.	0x00



Bits	Access	Name	Description	Reset
[11:0]	RO	fd_in_len	Length of the frame to be received in the RX queue	0x000

## UD\_GLB\_IQ\_ADDR

UD\_GLB\_IQ\_ADDR is an RX frame header address register.

The register does not support soft reset.

### NOTE

If the address assigned by software is not word aligned, the logic writes data according to the word aligned address. In this case, the previously written data is invalid. For example, if the configured header address of a frame is 0xF000\_8002 (non-word-aligned address), the logic writes 0x00 or other data to both the 0xF000\_8000 and 0xF000\_8001 addresses. Then, the logic writes the first byte (valid data) of the RX frame to the 0xF000\_8002 address, writes the second byte (valid data) of the RX frame to the 0xF000\_8003 address. Subsequent data is written to the buffer in sequence. If the configured header address of the RX frame is other non-word-aligned address, the logic writes data in a similar way.

Offset Address: 0x0358 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	startaddr_iq	Header address (configured by the CPU) of the storage space corresponding to the RX frame. The RX frame requests the bus according to this address.	0x00000000

## UD\_GLB\_BFC\_STAT

UD\_GLB\_BFC\_STAT is a counter for traffic control status of forward buffer and aging time of multi-packet interrupt.

The register does not support soft reset.

Offset Address: 0x035C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	timerover_cnt	Register for the count of multi-packet interrupt aging time counter overflow events (the count reaches the configured value)  NOTE  If the value of timerover_cnt is too large in a	0x0000



			unit of time, it indicates that <a href="#">UD_GLB_IRQN_SET[int_frm_cnt]</a> is set improperly. Multi-packet interrupt is triggered by the aging time. Therefore, the configured value must be reduced.	
[15:0]	RO	flowctrl_cnt	<p>Register for the count of the forward buffer of the uplink or downlink port entering the traffic control status</p> <p> <b>NOTE</b></p> <p>If the value of flowctrl_cnt is too large in a unit of time, it indicates that <a href="#">UD_GLB_FC_LEVEL[blimit_up]</a> or <a href="#">UD_GLB_FC_LEVEL[blimit_down]</a> is set to a too small value, or the external network condition is worsened. In this case, the configured value may be reduced.</p>	0x0000

## UD\_GLB\_EQ\_ADDR

UD\_GLB\_EQ\_ADDR is a TX queue header address register.

The register does not support soft reset.



If the header address of the TX frame is not word aligned, the logic reads data according to the word aligned address. In this case, the previously read data is invalid and discarded. For example, if the configured header address of the TX frame is 0xF000\_8102 (non-word-aligned address), the logic directly discards the byte data read from the 0xF000\_8100 and 0xF000\_8101 addresses. Then, the logic considers the data read from the 0xF000\_8102 address as the first byte (valid data) of the TX frame and considers the data read from the 0xF000\_8103 address as the second byte (valid data) of the TX frame. All subsequent data is valid (until the data of the specified frame length is read). If the configured header address of the TX frame is other non-word-aligned address, the logic reads data in a similar way.

Offset Address: 0x0360 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	add_fd_addr_out	Header address of the TX frame added by the CPU to the TX queue	0x00000000

## UD\_GLB\_EQFRM\_LEN

UD\_GLB\_EQFRM\_LEN is a TX queue frame length configuration register.

The register does not support soft reset.



Offset Address: 0x0364 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:11]	RO	reserved	Reserved	0x0000000
[10:0]	RW	add_fd_len_out	Length of the TX frame added by the CPU to the TX queue  Configure this register to trigger hardware so that software can write the header address and length of the TX frame to the TX queue for transmission. When transmitting a frame, software must write the header address of the frame before the length of the frame.  Note: The frames whose add_fd_len_out is less than 20 bytes or greater than 1600 bytes are discarded. In other words, the allowed range is from 20 bytes to 1600 bytes.	0x000

## UD\_GLB\_QSTAT

UD\_GLB\_QSTAT is a queue status register.

The register does not support soft reset.

Offset Address: 0x0368 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:30]	RO	reserved	Reserved	0x0
[29:24]	RO	iq_in_index	RX index of the RX (packet RX) queue	0x00
[23:22]	RO	reserved	Reserved	0x0
[21:16]	RO	cpuw_index	RX index of frame header address of the RX (packet RX) queue	0x00
[15:14]	RO	reserved	Reserved	0x0
[13:8]	RO	eq_in_index	RX index of frame descriptor of the TX (packet TX) queue	0x00
[7:6]	RO	reserved	Reserved	0x0
[5:0]	RO	eq_out_index	TX index of frame descriptor of the TX (packet TX) queue	0x00



## UD\_GLB\_ADDRQ\_STAT

UD\_GLB\_ADDRQ\_STAT is an address queue status register.

The register does not support soft reset.

Offset Address: 0x036C Total Reset Value: 0x0300\_0000

Bits	Access	Name	Description	Reset
[31:26]	RO	reserved	Reserved	0x00
[25]	RO	cpuaddr_in_rdy	Whether the CPU can configure the frame header address of the RX queue 0: The CPU cannot configure the frame header address of the RX queue. 1: The CPU can configure the frame header address of the RX queue.   <b>NOTE</b> The values of cpuaddr_in_rdy and eq_in_rdy are set to 0 during reset. The values, however, are set to 1 by the circuit immediately after reset. In other words, after reset, the iq address queue and eq descriptor queue are configurable.	0x1
[24]	RO	eq_in_rdy	Whether the CPU can configure the frame descriptor (header address and length) of the TX queue 0: The CPU cannot configure the frame descriptor (header address and length) of the TX queue. 1: The CPU can configure the frame descriptor (header address and length) of the TX queue.	0x1
[23:22]	RO	reserved	Reserved	0x0
[21:16]	RO	cpu_cnt	Header address count for available frames assigned by the CPU to the RX queue	0x00
[15:14]	RO	reserved	Reserved	0x0
[13:8]	RO	iq_cnt	Used length of the RX queue (0 to iq_len)	0x00
[7:6]	RO	reserved	Reserved	0x0
[5:0]	RO	eq_cnt	Used length of the TX queue (0 to eq_len)	0x00



## UD\_GLB\_FC\_TIMECTRL

UD\_GLB\_FC\_TIMECTRL is a traffic control time configuration register.

The register does not support soft reset.

Offset Address: 0x0370 Total Reset Value: 0x07FF\_86A0

Bits	Access	Name	Description	Reset
[31:27]	RO	reserved	Reserved	0x00
[26:17]	RW	flux_timer_cfg	Traffic limit time interval counter, which is used to count the frequency division clock generated by flux_timer_inter. If this counter is set to 0, traffic limit is not performed.	0x3FF
[16:0]	RW	flux_timer_inter	Traffic limit time granularity counter, which is used to count the ETH working clock (for details, see the PERI_CRG91 register). The default field value is 100000.  For example, if the default ETH working clock is 54 MHz and the time granularity is 1 ms, this field is set to 54000, which is calculated as follows: $1 \text{ ms}/(1/54 \text{ MHz}) = [1 \times 10^{(-3)}]/[(1/54) \times 10^{(-6)}] = 54 \times 10^3 = 54000$	0x186A0

## UD\_GLB\_FC\_RXLIMIT

UD\_GLB\_FC\_RXLIMIT is a traffic control limit configuration register.

The register does not support soft reset.

Offset Address: 0x0374 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:20]	RO	reserved	Reserved	0x000
[19:0]	RW	flux_cfg	Traffic limit upper threshold register. This group of bits is used to limit the number of frames received by software in the traffic limit time interval. The frames received after the configured upper	0x00000



Bits	Access	Name	Description	Reset
			threshold is exceeded are selectively discarded or received according to the configuration. When this group of bits is all set to 0, it indicates that traffic limit is not performed.	

## UD\_GLB\_FC\_DROPCTRL

UD\_GLB\_FC\_DROPCTRL is a packet drop control register for traffic limit.

The register does not support soft reset.

Offset Address: 0x0378 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:3]	RO	reserved	Reserved	0x00000000
[2]	RW	flux_uni	Whether unicast packets are discarded when the upper threshold of traffic limit is exceeded 0: Do not discard unicast packets. 1: Discard unicast packets.	0x0
[1]	RW	flux_multi	Whether multicast packets are discarded when the upper threshold of traffic limit is exceeded 0: Do not discard multicast packets. 1: Discard multicast packets.	0x0
[0]	RW	flux_broad	Whether broadcast packets are discarded when the upper threshold of traffic limit is exceeded 0: Do not discard broadcast packets. 1: Discard broadcast packets.	0x0

## UD\_GLB\_RX\_COE\_EN

UD\_GLB\_RX\_COE\_EN is an RX COE enable register.

The register does not support soft reset.

Offset Address: 0x0380 Total Reset Value: 0x8000\_E000



Bits	Access	Name	Description	Reset
[31]	RW	rx_coe_en	COE enable in the RX direction	0x1
[30:16]	RO	reserved	Reserved	0x0000
[15]	RW	iphdr_drop	Packet discarding enable when a checksum error occurs in the IPv4 IP header 0: The packet is transparently transmitted. 1: The packet is discarded.	0x1
[14]	RW	pro_drop	TCP/UDP packet discarding enable when a checksum error occurs 0: The packet is transparently transmitted. 1: The packet is discarded.	0x1
[13]	RW	i6udp_drop	Discarding enable for the invalid IPv6 UDP packet when the checksum domain is 0 0: The packet is transparently transmitted. 1: The packet is discarded.	0x1
[12:0]	RO	reserved	Reserved	0x0000

#### 5.1.5.4 Description of the Statistics Counter Control Registers

##### UD\_STS\_PORTCNT

UD\_STS\_PORTCNT is a port status counter.

The register does not support soft reset.

Offset Address: 0x0584 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:28]	RO	rxsof_cnt	Count of the frame headers received by the port	0x0
[27:24]	RO	rxeof_cnt	Count of the frame trailers received by the port	0x0
[23:20]	RO	rxcrcok_cnt	Count of the frames without CRC errors received by the port	0x0



Bits	Access	Name	Description	Reset
[19:16]	RO	rxcrcbad_cnt	Count of the frames with CRC errors received by the port	0x0
[15:12]	RO	txsof_cnt	Count of the frame headers transmitted by the port	0x0
[11:8]	RO	txeof_cnt	Count of the frame trailers transmitted by the port	0x0
[7:4]	RO	txcrcok_cnt	Count of the frames without CRC errors transmitted by the port	0x0
[3:0]	RO	txcrcbad_cnt	Count of the frames with CRC errors transmitted by the port	0x0

## UD\_PORT2CPU\_PKTS

UD\_PORT2CPU\_PKTS is a register for the total number of packets received by the CPU from the uplink or downlink port.

The register does not support soft reset.

Offset Address: 0x05A0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	Reserved	0x0000
[15:0]	WC	pkts_cpu	Total number of the packets received by the CPU port from the uplink or downlink port. Writing 0 clears this register. Writing 1 has no effect.	0x0000

## UD\_CPU2IQ\_ADDRCNT

UD\_CPU2IQ\_ADDRCNT is a register for the count of configuring packet receiving address queue by the CPU.

The register does not support soft reset.

Offset Address: 0x05A4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	Reserved	0x0000
[15:0]	WC	addr_cpu	Count of configuring packet receiving	0x0000



Bits	Access	Name	Description	Reset
			address queue by the CPU successfully. Writing 0 clears this register. Writing 1 has no effect.	

## UD\_RX\_IRQCNT

UD\_RX\_IRQCNT is a register for the count of reporting single-packet interrupt by the uplink or downlink port.

The register does not support soft reset.

Offset Address: 0x05A8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	Reserved	0x0000
[15:0]	WC	pkts_port	Count of the frame RX interrupts reported by the uplink or downlink port. Writing 0 clears this register. Writing 1 has no effect.	0x0000

## UD\_CPU2EQ\_PKTS

UD\_CPU2EQ\_PKTS is a register for the total number of packets transmitted by the CPU to the uplink or downlink port.

The register does not support soft reset.

Offset Address: 0x05AC Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	reserved	Reserved	0x0000
[15:0]	WC	pkts_cpu2tx	Total number of packets transmitted by the CPU to the uplink or downlink port. Writing 0 clears this register. Writing 1 has no effect.	0x0000

### 5.1.5.5 Description of the Statistics Result Registers

Statistics result registers can be configured in two modes: read only and read clear. If [UD\\_MAC\\_SET\[cntr\\_rdclr\\_en\]](#) is set to 1, it indicates the read clear mode. If



[UD\\_MAC\\_SET\[cntr\\_rdclr\\_en\]](#) is set to 0, it indicates the read only mode. The following registers are described only in read only mode.

## UD\_RX\_DVCNT

UD\_RX\_DVCNT is an RXDV rising edge count register.

The register does not support soft reset.

Offset Address: 0x0600 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	rxdvrise	Count of all RXDV rising edges	0x00000000

## UD\_RX\_OCTS

UD\_RX\_OCTS is a register for the total number of bytes received.

The register does not support soft reset.

Offset Address: 0x0604 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	ifinoctets	Count of all received bytes, including the bytes in correct frames, error frames, and preambles. The frames without valid start of frame delimiters (SFDs) are not counted.	0x00000000

## UD\_RX\_RIGHTOCTS

UD\_RX\_RIGHTOCTS is a register for the total number of bytes of received correct packets.

The register does not support soft reset.

Offset Address: 0x0608 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	octets_rx	Count of the received bytes, including the bytes in correct frames and error frames but excluding the bytes in preambles. The frames without valid SFDs are not counted.	0x00000000



## UD\_HOSTMAC\_PKTS

UD\_HOSTMAC\_PKTS is a register for the number of packets matching the local MAC address.

The register does not support soft reset.

Offset Address: 0x060C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	local_mac_match	Count of correct RX frames whose destination MAC address is the same as the local MAC address, excluding short frames, long frames, frames with CRC errors, pause frames, and error TX frames.	0x00000000

## UD\_RX\_RIGHTPKTS

UD\_RX\_RIGHTPKTS is a register for the total number of packets received by the port.

The register does not support soft reset.

Offset Address: 0x0610 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	pkts	Count of all frames	0x00000000

## UD\_RX\_BROADPKTS

UD\_RX\_BROADPKTS is a register for the number of correct broadcast packets.

The register does not support soft reset.

Offset Address: 0x0614 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	broadcastpkts	Count of broadcast frames with valid length and without CRC errors, excluding pause frames and error TX frames	0x00000000



## UD\_RX\_MULTPKTS

UD\_RX\_MULTPKTS is a register for the number of correct multicast packets.

The register does not support soft reset.

Offset Address: 0x0618 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	multicastpkts	Count of multicast frames with valid length and without CRC errors, excluding pause frames and error TX frames	0x00000000

## UD\_RX\_UNIPKTS

UD\_RX\_UNIPKTS is a register for the number of correct unicast packets.

The register does not support soft reset.

Offset Address: 0x061C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	ifinunicastpkts	Count of unicast frames with valid length and without CRC errors, excluding pause frames and error TX frames	0x00000000

## UD\_RX\_ERRPKTS

UD\_RX\_ERRPKTS is a register for the total number of incorrect packets.

The register does not support soft reset.

Offset Address: 0x0620 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	ifinerrors	Count of all error frames, including frames with CRC errors, short frames, long frames, and error TX frames	0x00000000

## UD\_RX\_CRCERR\_PKTS

UD\_RX\_CRCERR\_PKTS is a register for the count of CRC errors.

The register does not support soft reset.



Offset Address: 0x0624 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	crcerr	Count of RX frames with valid length (non short and long frames) but with CRC or alignment errors	0x00000000

## UD\_RX\_LENERR\_PKTS

UD\_RX\_LENERR\_PKTS is a register for the number of packets with invalid length.

The register does not support soft reset.

Offset Address: 0x0628 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	abnormalsizepkts	Count of frames (short frames and long frames) with invalid length (less than the set minimum valid length or greater than the set maximum valid length)	0x00000000

## UD\_RX\_OCRCERR\_PKTS

UD\_RX\_OCRCERR\_PKTS is a register for the number of packets with odd nibbles and CRC errors.

The register does not support soft reset.

Offset Address: 0x062C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3alignmenterr	Received frames with odd nibbles and CRC errors	0x00000000

## UD\_RX\_PAUSE\_PKTS

UD\_RX\_PAUSE\_PKTS is a register for the number of received pause packets.

The register does not support soft reset.

Offset Address: 0x0630 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RO	dot3pause	Count of received pause frames	0x00000000

## UD\_RF\_OVERCNT

UD\_RF\_OVERCNT is a register for the count of RXFIFO overflow events.

The register does not support soft reset.

Offset Address: 0x0634      Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dropevents	Accumulative count of RXFIFO overflow events during the reception of frames	0x00000000

## UD\_FLUX\_TOL\_IPKTS

UD\_FLUX\_TOL\_IPKTS is a register for the total number of received packets allowed by the traffic limit.

The register does not support soft reset.

Offset Address: 0x0638      Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	flux_frame_cnt	Total count of correct RX frames allowed by the traffic limit, excluding short frames, long frames, frames with CRC errors, pause frames, and error TX frames	0x00000000

## UD\_FLUX\_TOL\_DPKTS

UD\_FLUX\_TOL\_DPKTS is a register for the total number of packets discarded due to traffic limit. The register does not support soft reset.

Offset Address: 0x063C      Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	flux_drop_cnt	Count of correct frames discarded due to traffic limit, excluding short frames, long frames, frames with CRC errors, pause frames, and error TX frames	0x00000000



## UD\_MN2CPU\_PKTS

UD\_MN2CPU\_PKTS is a register for the number of packets not forwarded to the CPU port due to MAC limit. The register does not support soft reset.

Offset Address: 0x064C      Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	mac_not2cpu_pkts	Number of packets not forwarded to the CPU port due to MAC limit	0x00000000

## UD\_TX\_PKTS

UD\_TX\_PKTS is a register for the total number of packets transmitted successfully. The register does not support soft reset.

Offset Address: 0x0780      Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	pkts_tx	Count of all configured TX frames, excluding the frames discarded due to timeout and the TX frames whose length of <a href="#">UD_GLB_EQFRM_LEN</a> is not within valid range	0x00000000

## UD\_TX\_BROADPKTS

UD\_TX\_BROADPKTS is a register for the number of broadcast packets transmitted successfully. The register does not support soft reset.

Offset Address: 0x0784      Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	broadcastpkts_tx	Count of broadcast frames transmitted successfully (excluding retransmission)	0x00000000

## UD\_TX\_MULTPKTS

UD\_TX\_MULTPKTS is a register for the number of multicast packets transmitted successfully. The register does not support soft reset.



Offset Address: 0x0788 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	multicastpkts_tx	Count of multicast frames transmitted successfully (excluding retransmission)	0x00000000

## UD\_TX\_UNIPKTS

UD\_TX\_UNIPKTS is a register for the number of unicast packets transmitted successfully. The register does not support soft reset.

Offset Address: 0x078C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	ifoutucastpkts_tx	Count of unicast frames transmitted successfully (excluding retransmission)	0x00000000

## UD\_TX\_OCTS

UD\_TX\_OCTS is a register for the total number of transmitted bytes. The register does not support soft reset.

Offset Address: 0x0790 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	octets_tx	Total count of transmitted bytes, including the bytes of retransmit frames, correct frames, and error frames, but excluding the preamble bytes	0x00000000

## UD\_TX\_PAUSE\_PKTS

UD\_TX\_PAUSE\_PKTS is a register for the number of transmitted pause frames. The register does not support soft reset.

Offset Address: 0x0794 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3outpause	Count of transmitted pause frames	0x00000000



## UD\_TX\_RETRYCNT

UD\_TX\_RETRYCNT is a register for the total count of retransmission. The register does not support soft reset.

Offset Address: 0x0798 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	retry_times_tx	Total count of retransmissions of TX frames	0x00000000

## UD\_TX\_COLCNT

UD\_TX\_COLCNT is a register for the total count of collisions. The register does not support soft reset.

Offset Address: 0x079C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	collisions	Count of collisions	0x00000000

## UD\_TX\_LC\_PKTS

UD\_TX\_LC\_PKTS is a register for the number of packets with late collision. The register does not support soft reset.

Offset Address: 0x07A0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3latecol	Count of packets with late collision	0x00000000

## UD\_TX\_COLOK\_PKTS

UD\_TX\_COLOK\_PKTS is a register for the number of packets transmitted successfully with collisions. The register does not support soft reset.

Offset Address: 0x07A4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3col_ok	Count of packets transmitted successfully with collisions	0x00000000



## UD\_TX\_RETRY15\_PKTS

UD\_TX\_RETRY15\_PKTS is a register for the number of packets discarded due to more than 15 times of retransmission. The register does not support soft reset.

Offset Address: 0x07A8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3excessivecol	Count of the packets discarded due to more than 15 times of retransmission	0x00000000

## UD\_TX\_RETRYN\_PKTS

UD\_TX\_RETRYN\_PKTS is a register for the number of packets with the count of collisions being equal to the threshold. The register does not support soft reset.

Offset Address: 0x07AC Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	dot3colcnt	Count of packets with the count of collisions being equal to the threshold. This register is set by <a href="#">UD_MAC_SET</a> [colthreshold]	0x00000000

## 5.2 FE PHY

### 5.2.1 Overview

The fast Ethernet physical layer (FE PHY) module supports single-port 10BASE-T, 10BASE-Te, and 100BASE-TX.

### 5.2.2 Features

The FE PHY realizes 10BASE-T, 10BASE-Te, and 100BASE-TX at the Ethernet physical layer. At the cable end, the medium dependent interface (MDI) of the FE PHY module directly connects to the twisted pair through the transformer and RJ45 interface. At the media access controller (MAC) end, the MII of this module connects to the MAC layer.

The FE PHY module supports the following features:



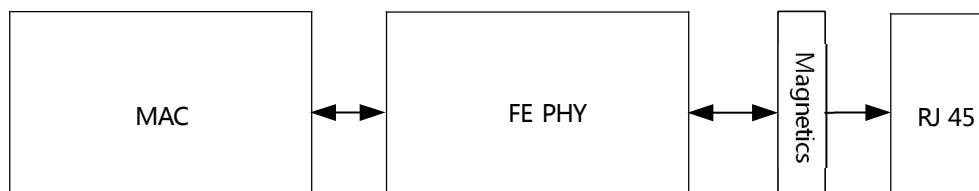
- Compatibility with Institute of Electrical and Electronics Engineers (IEEE) 10BASE-T and 100BASE-TX
- Cable status diagnosis
- MDIO interface and MII
- Auto-negotiation and parallel detection of IEEE802.3u
- Automatic crossover, switchover, and polarity adjustment of a cable pair
- 802.3az (Energy Efficient Ethernet)
- Direct transmission of the eye pattern test sequence at 10 Mbit/s or 100 Mbit/s by configuring the register, independent from the Tesgine
- Voltage-type output, with transformer tap connecting to the GND through a capacitor and no pull-up and pull-down resistors

### 5.2.3 Typical Application

[Figure 5-5](#) shows the typical application of the FE PHY.

- At the cable end, the MDI directly connects to the twisted pair through a transformer and RJ45 interface.
- At the MAC end, the FE PHY connects to the MAC layer through the MII interface.

**Figure 5-5** Typical application of the FE PHY



For recommendations on the peripheral design and power supply mode, see FE PHY-related descriptions in the *Hi3516CV610 Hardware Design User Guide*.

**Table 5-8** MDI pin description

Pin Name	I/O	Attribute	Function Description
ETH_MDI_AN	IO	Analog	Differential RX and TX signals
ETH_MDI_AP	IO	Analog	The mode can be forcibly or adaptively set to 10BASE-T or 100BASE-TX. The MDI and MDI-X interconnection modes are supported. The polarity of MDI_N and MDI_P can be
ETH_MDI_BN	IO	Analog	



Pin Name	I/O	Attribute	Function Description
ETH_MDI_BP	IO	Analog	<p>corrected.</p> <p>Note:</p> <p>When the automatic medium dependent interface crossover (MDIX) function is used in 10BASE-T, 10BASE-Te, and 100BASE-TX mode, the TX channel of the peer end is detected, and TX and RX channels are configured.</p>

## 5.2.4 Configuration

This section describes the configuration modes supported by the FE PHY core.

Related terms are as follows:

- Manual mode configuration
- Speed/Duplex selection
- Auto-negotiation
- Priority solution
- Parallel detection (non-auto-negotiation)
- Restart auto-negotiation
- AUTO-MDIX configuration
- Auto polarity correction
- PHY address
- Low power mode
- Expand register

The FE PHY core supports the following four Ethernet protocols:

- 10BASE-T Full Duplex
- 10BASE-T Half Duplex
- 100BASE-TX Full Duplex
- 100BASE-TX Half Duplex

You can select the speed and duplex mode by configuring the register:

- Manual configuration
- Auto-negotiation

### 5.2.4.1 Manual Configuration

The FE PHY core can be configured in multiple modes. All configurations can be managed by reading and writing registers over the MDIO interface. To use the



manual configuration mode, disable auto-negotiation first. The steps to use the mode are as follows:

- Step 1** Configure bit 12 of register BMCR (offset address: 0x00) as 0x0.
- Step 2** Configure SPEED[1:0] of register BMCR (offset address: 0x00) to determine the speed.
- Step 3** Configure Duplex[1:0] of register BMCR (offset address: 0x00) to determine the duplex mode.

----End

#### 5.2.4.2 Auto-Negotiation

The FE PHY in 10BASE-T, 10BASE-Te, and 100BASE-TX mode can be set to support auto-negotiation or not. When the peer-end device does not support auto-negotiation, determine the connection mode by parallel detection.

#### Speed and Duplex Selection and Priority Solution

Auto-negotiation provides a mechanism in which interconnected parties can exchange the configuration information. This mechanism is implemented by exchanging fast link pulses (FLPs). During the auto-negotiation, the protocol that is supported by both parties and has the best performance will be preferentially selected.

Priorities of the modes during auto-negotiation selection are sequenced as follows:

1. 100BASE-TX Full Duplex (highest priority)
2. 100BASE-TX Half Duplex
3. 10BASE-T Full Duplex
4. 10BASE-T Half Duplex (lowest priority)

#### Parallel Detection

If the peer-end device does not support auto-negotiation, the auto-negotiation module determines the connection protocol to be used based on the received information. (10BASE-T and 100BASE-X send different signals to indicate the link status.) If FE PHY uses parallel detection to implement the auto-negotiation process, the following registers can be used to determine the current status:

- Bits 5 and 7 of register ANLPARD indicate the peer connection mode.
- If bit[4:0] of register ANLPARD are 0x00001, the register indicates parallel detection success and a connected device that complies with IEEE802.3.
- If bit 0 of register ANER (offset address: 0x06) is 0, the negotiation is implemented by parallel detection.
- If bit 4 of register ANER (offset address: 0x06) is 1, parallel detection errors occur and the link is not set up.



## Restart Auto-Negotiation

If a link is up and then down after an auto-negotiation is set up successfully, you can restart auto-negotiation to configure a link again.

- If auto-negotiation is enabled, write 0x1 to bit 9 of register BMCR (offset address: 0x00) to restart auto-negotiation.
- If auto-negotiation is disabled, write 0x1 to bit 12 of register BMCR (offset address: 0x00) to enable auto-negotiation.

## 5.2.4.3 AUTO—MDIX

The FE PHY core can detect whether the peer PHY is connected through a crossover cable or a straight-through cable. It automatically adjusts channels A and B to initialize connection in 10BASE-T and 100BASE-TX modes. The AUTO-MDIX interface is adjusted before the auto-negotiation information is interacted. You can confirm the current status by reading the following registers:

- Bit 10 of register LINK\_AN (offset address: 0x11) indicates the MDIX status.
- Bit 15 of register AUX\_CTL (offset address: 0x12) indicates AUTO-MDIX enabling status.
- If AUTO-MDIX is disabled, configure bit 14 of register AUX\_CTL (offset address: 0x12) to use the MDI or MDIX mode.

## 5.2.4.4 Polarity Correction

The FE PHY core can automatically detect and correct the polarity of the RX twisted pair.

Bit[15:12] of register LINK\_AN (offset address: 0x11) indicates the polarity of the RX twisted pair.

## 5.2.4.5 PHY Address

The 5-bit PHY address of the FE PHY core is unique and can be changed in the following two modes:

- Change during power-on initialization  
Configure PERI\_FEPHY bit[4:0] before the PHY is powered on or reset so that the PHY implements sampling during the initialization.
- Online change  
Configure bit[4:0] of register PHY ADDRESS (offset address: 0x1f) after the PHY is powered on to redefine the PHY address.

## 5.2.4.6 MII Isolate

In the Isolate mode, the auto-negotiation function of the FE PHY core is disabled but you can still control the interface working mode through the MDIO interface by performing the following steps:



**Step 1** Configure bit 10 of register BMCR (offset address: 0x00) as 0x1 to enter the Isolate mode.

**Step 2** Configure SPEED[1:0] of register BMCR (offset address: 0x00) to control the current mode.

----End

#### 5.2.4.7 Power Down Mode

When FE PHY enters the power down mode, internal circuits are powered off except the management interfaces (MDC and MDIO). In this mode, the power consumption of the FE PHY core is minimum.

Configure bit 11 of register BMCR (offset address: 0x00) as 0x1 to enter the power down mode.

#### 5.2.4.8 Expand Register Access

##### Accessing a Dedicated Expand Register

The FE PHY core contains expand registers besides the 32 basic registers defined by the IEEE. The expand registers are placed in the expand register space and should be accessed in the expanded addressing mode of basic registers. To access an expand register in this mode, perform the following steps:

**Step 1** Configure the PHY to enter the power down mode so that internal actions do not affect register operations.

**Step 2** Write the address of the register to be accessed to EXPMA (offset address: 0x1e). To write the register, go to [Step 3](#). To read the register, go to [Step 4](#).

**Step 3** Write the required data to register EXPMD (offset address: 0x1d).

**Step 4** Read register EXPMD (offset address: 0x1d).

**Step 5** Configure the PHY to exit the power down mode.

----End

##### Accessing a Clause22 Expand Register

The Clause22 standard defines an expansion mode, which is implemented by registers MACR (offset address: 0xD) and MAADR (offset address: 0xE).

To access an expand register in this mode, perform the following steps:

**Step 1** Configure the function field of register MACR as 0x00 to select the address. Then, write the address of the MMD you want to operate to the DEVAD field.

**Step 2** Write the address of the MMD register you want to operate to MAADR.



**Step 3** Configure the function field of register MACR as 0x00 to select data, no post increment. Write the address of the MMD you want to operate to the DEVAD field. To write the MMD, go to [Step 4](#). To read the MMD, go to [Step 5](#).

**Step 4** Write the required data to MAADR.

**Step 5** Read MAADR to obtain the required data.

----End

## 5.2.5 Summary of FEPHY\_BASE Registers

[Table 5-9](#) describes the FEPHY\_BASE registers.

### NOTE

The registers can be accessed through the MDIO interface.

**Table 5-9** Summary of the FEPHY\_BASE registers

Offset Address	Register	Description	Page
0x00	BMCR	Basic mode control register	<a href="#">5-70</a>
0x01	BMSR	IEEE basic MII status register	<a href="#">5-72</a>
0x02	OUI_1	IEEE PHY identifier 1 register	<a href="#">5-74</a>
0x03	OUI_2	IEEE PHY identifier 2 register	<a href="#">5-74</a>
0x04	ANAR	IEEE auto-negotiation TX register	<a href="#">5-75</a>
0x05	ANLPAR	IEEE auto-negotiation RX register	<a href="#">5-76</a>
0x06	ANER	IEEE auto-negotiation expand register	<a href="#">5-77</a>
0x0D	MACR	Expand control register	<a href="#">5-78</a>
0x0E	MAADR	Expand address data register	<a href="#">5-79</a>
0x11	LINK_AN_SR	Connection and auto-negotiation status register	<a href="#">5-79</a>
0x12	AUX_CTL	Auxiliary control register	<a href="#">5-80</a>
0x1D	EXPMD	Expand access data register	<a href="#">5-81</a>
0x1E	EXPMA	Expand access address register	<a href="#">5-81</a>
0x1F	PHY_ADDRESS	Broadcast address register	<a href="#">5-82</a>



## 5.2.6 Description of FEPHY\_BASE Registers

### BMCR

BMCR is a basic mode control register.

Offset Address: 0x00 Total Reset Value: 0x3100

Bits	Access	Name	Description	Reset
[15]	RWSC	mr_reset	<p>Reset 0: normal working mode 1: initialization soft reset/reset during working</p> <p> <b>NOTE</b> This bit resets the PHY status and control register to the default status. This bit is automatically cleared. When reset is not finished, the returned value remains 1 (with the duration of about 1.2 ms). Once auto-negotiation starts or the device enters the forcible mode, reset finishes.</p>	0x0
[14]	RW	mr_loopback	<p>Loopback enable 0: normal working mode 1: enabled The loopback function transmits the data sent by MII/GMII to the data RX path of MII/GMII. Data is transmitted circularly between the DAC/ADC subsystem of the TX and RX modules. The loopback mode can detect most signal paths in the PHY of the transmitter and receiver.</p>	0x0
[13]	RW	mr_speed_0	<p>Speed selection When auto-negotiation is disabled, mr_speed_0 forcibly sets the device speed as follows: mr_speed_0 [1:0] 00: 10 Mbit/s 01: 100 Mbit/s 10: reserved 11: reserved</p> <p> <b>NOTE</b> When auto-negotiation is enabled, this bit</p>	0x1



Bits	Access	Name	Description	Reset
			does not function. mr_speed_0[1:0] consists of HA_CONTROL[6] and LA_CONTROL[5].	
[12]	RW	mr_an_enable	Auto-negotiation enable 0: disabled 1: enabled	0x1
[11]	RW	mr_power_down	Low power mode enable 0: normal working mode 1: low power mode (Only the MDIO interface functions normally.)	0x0
[10]	RW	mr_isolate	Isolation enable 0: normal working mode 1: isolation enabled. The GPHY core does not respond to the MAC data transmission. The isolation mode sets the PHY to the low power mode and all media operations are interrupted.	0x0
[9]	RWSC	mr_restart_an	Restart auto-negotiation enable 0: normal working mode 1: Restart and initialize auto-negotiation. If auto-negotiation is disabled, this bit does not function. After auto-negotiation is started, this bit is automatically cleared. The returned value remains 1 during the initialization. Clearing this bit does not affect auto-negotiation.	0x0
[8]	RW	mr_duplex	Duplex mode 0: half duplex 1: full duplex  <b>NOTE</b> When auto-negotiation is enabled, this bit is invalid.	0x1
[7]	RW	mr_col_test	Conflict test 0: normal working mode 1: conflict test enabled	0x0
[6]	RW	mr_speed_1	Speed selection When auto-negotiation is disabled, mr_speed_1 forcibly sets the device speed	0x0



Bits	Access	Name	Description	Reset
			<p>as follows:</p> <p>mr_speed_1[1:0]</p> <p>00: 10 Mbit/s</p> <p>01: 100 Mbit/s</p> <p>10: reserved</p> <p>11: reserved</p> <p> <b>NOTE</b></p> <p>When auto-negotiation is enabled, this bit does not function. mr_speed_1[1:0] consists of HA_CONTROL[6] and LA_CONTROL[5].</p>	
[5:0]	RW	ieee_reserved	<p>Reserved. Writing this field has no effect and reading this field returns 0.</p> <p> <b>NOTE</b></p> <p>For all reserved bits, writing has no effect and the read value is 0.</p>	0x00

## BMSR

BMSR is an IEEE basic MII status register.

Offset Address: 0x01 Total Reset Value: 0x7849

Bits	Access	Name	Description	Reset
[15]	RO	base_t4_100	<p>Whether the 100BASE-T4 mode is supported</p> <p>0: not supported</p> <p>1: supported</p> <p> <b>NOTE</b></p> <p>When this bit is set to 1, the chip supports the 100BASE-T4 mode. When this bit is set to 0, the chip does not support this mode.</p>	0x0
[14]	RO	base_xfd_100	<p>Whether the full-duplex 100BASE-TX mode is supported</p> <p>0: not supported</p> <p>1: supported</p>	0x1
[13]	RO	base_xhd_100	<p>Whether the half-duplex 100BASE-TX mode is supported</p>	0x1



Bits	Access	Name	Description	Reset
			0: not supported 1: supported	
[12]	RO	base_fd_10	Whether the full-duplex 10BASE-T mode is supported 0: not supported 1: supported	0x1
[11]	RO	base_hd_10	Whether the half-duplex 10BASE-T mode is supported 0: not supported 1: supported	0x1
[10]	RO	base_t2fd_100	Whether the full-duplex 10BASE-T2 mode is supported 0: not supported 1: supported	0x0
[9]	RO	base_t2hd_100	Whether the half-duplex 10BASE-T2 mode is supported 0: not supported 1: supported	0x0
[8]	RO	extended_status	Whether the 1000BASE-T expand status register is supported 0: not supported 1: supported	0x0
[7]	RO	reserved	Reserved	0x0
[6]	RO	mr_preamble_suppress	Preamble limitation enable 1: After the PHY is reset, only one 32-bit preamble is required for normal transmission.	0x1
[5]	RO	mr_an_complete	Auto-negotiation completion flag 0: not complete 1: complete	0x0
[4]	RO	mr_rem_fault	Remote-end error detection indicator 0: no remote-end error detected 1: remote-end error detected  <b>NOTE</b> Reading or resetting clears this bit.	0x0



Bits	Access	Name	Description	Reset
[3]	RO	mr_an_capable	Auto-negotiation enabling allowed or not 0: not allowed 1: allowed	0x1
[2]	RO	mr_link_up	Connection status indicator 0: connection error since the register is read last time 1: normal connection since the register is read last time <b>NOTE</b> This bit is cleared in the case of disconnection. If the connection is normal when the register is read, this bit is set to 1.	0x0
[1]	RO	mr_jabber	Jabber detection 0: jabber conflict not detected 1: jabber conflict detected <b>NOTE</b> When jabber is detected in the 10BASE-T mode, this bit is set to 1.	0x0
[0]	RO	mr_extended_status	1000BASE-T expand status register enable 0: disabled 1: enabled	0x1

## OUI\_1

OUI\_1 is an IEEE PHY identifier 1 register.

Offset Address: 0x02 Total Reset Value: 0x2066

Bits	Access	Name	Description	Reset
[15:0]	RO	oui_msb	Upper 16 bits of the unique official identifier	0x2066

## OUI\_2

OUI\_2 is an IEEE PHY identifier 2 register.



Offset Address: 0x03 Total Reset Value: 0x9903

Bits	Access	Name	Description	Reset
[15:10]	RO	oui_lsb	Lower 16 bits of the unique official identifier	0x26
[9:4]	RO	mmn	Manufacturer model	0x10
[3:0]	RO	revision	Version number	0x3

## ANAR

ANAR is an IEEE auto-negotiation TX register.

Offset Address: 0x04 Total Reset Value: 0x01E1

Bits	Access	Name	Description	Reset
[15]	RW	mr_an_adv_np	Next-page transmission indicator 0: next-page transmission not supported 1: next-page transmission supported	0x0
[14]	-	reserved	Reserved	0x0
[13]	RW	mr_an_adv_rf	Remote-end error flag 0: remote error not detected 1: remote error detected	0x0
[12:11]	-	reserved	Reserved	0x0
[10]	RW	mr_pause	Stop control 0: MAC/controller frame transmission interrupt supported 1: not supported	0x0
[9]	RW	mr_t4_ability	Whether the 100BASE-T4 mode is supported 0: not supported 1: supported	0x0
[8]	RW	mr_tech_ability_10_0_fd	Whether the full-duplex 100BASE-TX mode is supported 0: not supported 1: supported	0x1
[7]	RW	mr_tech_ability_10_0_hd	Whether the half-duplex 100BASE-TX mode is supported	0x1



Bits	Access	Name	Description	Reset
			0: not supported 1: supported	
[6]	RW	mr_tech_ability_10_fd	Whether the full-duplex 10BASE-T mode is supported 0: not supported 1: supported	0x1
[5]	RW	mr_tech_ability_10_hd	Whether the half-duplex 10BASE-T mode is supported 0: not supported 1: supported	0x1
[4:0]	RW	mr_selector_filed	Protocol selection field defined by IEEE 802.3. The value 0x01 indicates that the chip supports IEEE 802.3.	0x01

## ANLPAR

ANLPAR is an IEEE auto-negotiation RX register.

Offset Address: 0x05 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	RO	next_page	Next-page transmission indicator 0: The link partner does not support next-page transmission. 1: The link partner supports next-page transmission.	0x0
[14]	RO	acknowledge	Remote-end RX response flag 0: The link partner does not respond to RX. 1: The link partner responds to RX.	0x0
[13]	RO	remote_fault	Link partner remote-end error flag 0: remote error not detected 1: remote error detected	0x0
[12]	-	reserved	Reserved	0x0
[11]	RO	asymetricalpause	Whether the link partner supports asymmetrical pause	0x0



Bits	Access	Name	Description	Reset
			0: not supported 1: supported	
[10]	RO	pause	Whether the link partner supports frame pause 0: not supported 1: supported	0x0
[9]	RO	lp_base_t4_100	Whether the link partner supports 100BASE-T4 0: not supported 1: supported	0x0
[8]	RO	base_tx_fd_100	Whether the link partner supports full-duplex 100BASE-TX 0: not supported 1: supported	0x0
[7]	RO	base_tx_100	Whether the link partner supports half-duplex 100BASE-TX 0: not supported 1: supported	0x0
[6]	RO	base_tx_fd_10	Whether the link partner supports full-duplex 10BASE-TX 0: not supported 1: supported	0x0
[5]	RO	base_tx_10	Whether the link partner supports half-duplex 10BASE-TX 0: not supported 1: supported	0x0
[4:0]	RO	lp_selector_filed	Link partner selection field defined by IEEE 802.3. The value 0x01 indicates that the chip supports IEEE 802.3.	0x00

## ANER

ANER is an IEEE auto-negotiation expand register.

Offset Address: 0x06 Total Reset Value: 0x0004



Bits	Access	Name	Description	Reset
[15:5]	-	reserved	Reserved	0x000
[4]	RO	mr_exp_pdf	Parallel error detection 0: no error detected 1: error detected	0x0
[3]	RO	mr_exp_lp_np_abl_e	Whether the link partner supports next-page negotiation 0: not supported 1: supported	0x0
[2]	RO	mr_exp_np_able	Whether next-page negotiation is supported 0: not supported 1: supported	0x1
[1]	RO	mr_exp_page_rx	Whether connection codeword is received 0: not received 1: received	0x0
[0]	RO	mr_exp_la_an_enable	Whether the link partner supports auto-negotiation 0: not supported 1: supported	0x0

## MACR

MACR is an expand control register.

Offset Address: 0x0D    Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:14]	RW	function	Expand register function field 00: MAADR register address operation 01: MAADR register data operation Other values: reserved	0x0
[13:5]	-	reserved	Reserved	0x000
[4:0]	RW	devad	Device address	0x0



## MAADR

MAADR is an expand address data register.

Offset Address: 0x0E Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	address_data	Device address or data register	0x0000

## LINK\_AN\_SR

LINK\_AN\_SR is a connection and auto-negotiation status register.

Offset Address: 0x11 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:12]	RO	tp_polarity_3_0	Bits 7-4 indicate the polarity of cable pairs A to D, respectively. 0: normal 1: inverted	0x0
[11]	RO	power_down_status	Current power-on or power-off status of the port 0: non-power-off status 1: power-off status	0x0
[10]	RO	mdix_status	Internal cross over status 0: no internal cross over 1: internal cross over activated	0x0
[9]	RO	fifo_error	Whether overflow or underrun occurs in the TX first-in first-out (FIFO) 0: Overflow or underrun does not occur. 1: Overflow or underrun occurs.	0x0
[8]	RO	power_on_init_in_progress	Power-on process completion flag 0: complete 1: not complete	0x0
[7]	RO	shallow_loopback_status	Shallow loopback enable status 0: disabled 1: enabled	0x0
[6]	RO	deep_loopback_status	Deep loopback enable status	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[5]	RO	non_compliant_mode_status	Current compliant mode 1: 1000-BASE non-compliant mode 0: IEEE-compliant mode	0x0
[4:3]	RO	speed_status	Real-time speed display Speed[1:0] 00: 10 Mbit/s 01: 100 Mbit/s 10: reserved 11: reserved	0x0
[2]	RO	link_status	Real-time connection display 0: disconnected 1: connected	0x0
[1]	RO	duplex_status	Real-time duplex status 0: half duplex 1: full duplex	0x0
[0]	RO	config_status	Master-slave status 0: slave status 1: master status	0x0

## AUX\_CTL

AUX\_CTL is an auxiliary control register.

Offset Address: 0x12 Total Reset Value: 0x9000

Bits	Access	Name	Description	Reset
[15]	RW	auto_mdix_enable	MDIX mode selection 0: manual MDIX mode 1: automatic MDIX mode	0x1
[14]	RW	mdix_manual	Straight-through or crossover cable 0: straight-through cable 1: crossover cable	0x0



Bits	Access	Name	Description	Reset
[13]	-	reserved	Reserved	0x0
[12]	RW	loopback_10b_disable	10BASE half-duplex mode loopback enable 0: enabled 1: disabled	0x1
[11]	RW	force100	Forcible 100BASE-TX connection mode 0: normal mode 1: forcible 100BASE-TX connection mode	0x0
[10]	RW	force10	Forcible 10BASE-T or 10BASE-Te connection mode 0: normal mode 1: forcible 10BASE-T or 10BASE-Te connection mode	0x0
[9]	RW	reserved	Reserved	0x0
[8]	RW	en_ext_loop_detect	External loopback detection enable 0: disabled 1: enabled	0x0
[7:0]	-	reserved	Reserved	0x00

## EXPMD

EXPMD is an expand access data register.

Offset Address: 0x1D Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	exp_data_msb	Expand access data register	0x0000

## EXPMA

EXPMA is an expand access address register.

Offset Address: 0x1E Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	exp_addr_msb	Expand access address register	0x0000



## PHY\_ADDRESS

PHY\_ADDRESS is a broadcast address register.

Offset Address: 0x1F Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:13]	-	reserved	Reserved	0x0
[12:8]	RW	broadcast_address	Broadcast access address when broadcast is enabled	0x00
[7:5]	-	reserved	Reserved	0x0
[4:0]	RW	phyaddress	Address of the port accessed by the SM	0x00



# Contents

<b>6 Video Encoder .....</b>	<b>6-1</b>
6.1 Overview.....	6-1
6.2 VEDU .....	6-1
6.2.1 Overview.....	6-1
6.2.2 Features .....	6-1
6.2.3 Function Description .....	6-3
6.3 JPGE .....	6-5
6.3.1 Overview.....	6-5
6.3.2 Features .....	6-5
6.3.3 Function Description .....	6-5



## Figures

---

<b>Figure 6-1</b> Encoding functional block diagram of the VEDU .....	6-4
<b>Figure 6-2</b> Encoding functional block diagram of the JPGE .....	6-6



# 6 Video Encoder

## 6.1 Overview

The video encoder supports multiple protocols, including SVAC3.0, H.265, H.264, JPEG, and MJPEG. It consists of the video encoding/decoding unit (VEDU) and JPEG encoder (JPGE). The VEDU implements SVAC3.0, H.265, and H.264 encoding, and the JPGE implements JPEG and MJPEG encoding.

### NOTE

- For Hi3516CV610-00B/20B, only H.265 and H.264 are supported.
- For Hi3516CV610-00S/20S/10B/00G/20G, SVAC 3.0, H.265, and H.264 are supported. For Hi3516CV610-10B, SVAC 3.0 does not support SmartCRR.

## 6.2 VEDU

### 6.2.1 Overview

The VEDU is a CODEC that supports SVAC3.0, H.265 and H.264 protocol and performs encoding by using hardware. It features low CPU usage, low bus bandwidth, short delay, and low power consumption.

### 6.2.2 Features

The VEDU has the following features:

- Supports SVAC3.0 Main Profile, Level 8.0.30 encoding.
  - Motion compensation with 1/2 or 1/4 pixel precision
  - I frames and p frames
  - Up to two reference frames and long-term reference frames
  - Three prediction unit (PU) types of 32x32, 16x16, and 8x8 for inter-prediction
  - Four PU types of 32x32, 16x16, 8x8, and 4x4 for intra-prediction



- Skip mode and merge mode with a maximum of two candidate points to be merged
- Four transform unit (TU) types of 32x32, 16x16, 8x8, and 4x4
- CABAC entropy encoding
- De-blocking filtering
- Patch
- PRDO
- Sample adaptive offset (SAO)
- QPMap/SkipMap
- Supports ITU-T H.265 Main Profile@Level 5 Main Tier encoding.
  - Motion compensation with 1/2 or 1/4 pixel precision
  - I frames and p frames
  - Up to two reference frames and long-term reference frames
  - Three PU types of 32x32, 16x16, and 8x8 for inter-prediction
  - Four PU types of 32x32, 16x16, 8x8, and 4x4 for intra-prediction
  - Skip mode and merge mode with a maximum of two candidate points to be merged
  - Four transform unit (TU) types of 32x32, 16x16, 8x8, and 4x4
  - CABAC entropy encoding
  - De-blocking filtering
  - Sample adaptive offset (SAO)
  - QPMap/SkipMap
- Supports ITU-T H.264 High Profile/Main Profile/Baseline Profile@Level 5.1 encoding.
  - Motion compensation with 1/2 or 1/4 pixel precision
  - I frames and p frames
  - Two reference frames at most and long-term reference frames
  - Two PU types of 16x16, and 8x8 for inter-prediction
  - All PU types of 16x16, 8x8, and 4x4 for intra-prediction
  - Trans4x4 and trans8x8
  - CABAC and CAVLC entropy encoding
  - De-blocking filtering
- Supports H.264 scalable video coding (SVC) time-domain layering (SVC-T).
- Supports the following input image format: Semi-Planar YUV420
- Supports SVAC3.0/H.265/H.264 multi-stream encoding:
  - 3200 x 1800@30 fps + 1280 x 720@30 fps encoding
  - 3840 x 2160@20 fps + 1280 x 720@20 fps encoding
- Supports configurable image resolutions.



- Minimum image resolution: 114 x 114
- Maximum image resolution: 3840 x 2160
- Step of the image width or height: 2
- Supports region of interest (ROI) encoding.
  - A maximum of eight ROIs
  - Independent enable/disable control for the encoding function of each ROI
- Supports on-screen display (OSD) encoding protection that can be enabled or disabled.
- Supports OSD front-end overlaying.
  - OSD overlaying before encoding for a maximum of eight regions
  - OSD overlapping with the maximum size of the source image and within the image position range
  - 256-level alpha blending
  - OSD overlaying enable control
  - OSD ARGB1555, ARGB4444, CLUT2, and CLUT4 formats
- Supports color-to-gray encoding.
- Supports the low-delay input function.
- Supports the low-delay wrapping function.
- Supports input cropping.
- Supports mosaic overlaying at the video front end.
- Supports intelligent encoding 2.0.
- Supports CBR, VBR, ABR, AVBR, CVBR, QVBR, FixQp, and QpMap bit rate control modes.
- Supports a maximum output bit rate of 80 Mbit/s for SVAC3.0, H.265, and H.264.

#### NOTICE

The VEDU encoding protection function applies only to the OSD overlaid on the VEDU.

### 6.2.3 Function Description

[Figure 6-1](#) shows the functional block diagram of the VEDU.

Based on related protocols and algorithms, the VEDU supports motion estimation, inter-prediction, intra-prediction, motion vector prediction, transform/quantization, inverse transform/inverse quantization, entropy encoding, stream generation, de-blocking filtering, and SAO (SVAC3.0 and H.265). The ARM software controls the bit rate and handles interrupts.

Before the VEDU is enabled for video encoding, software allocates three types of buffers for the VEDU in the external DDR SDRAM:

- Input image buffer

The VEDU reads the source images to be encoded from this buffer during encoding. This buffer is written by the VICAP module.

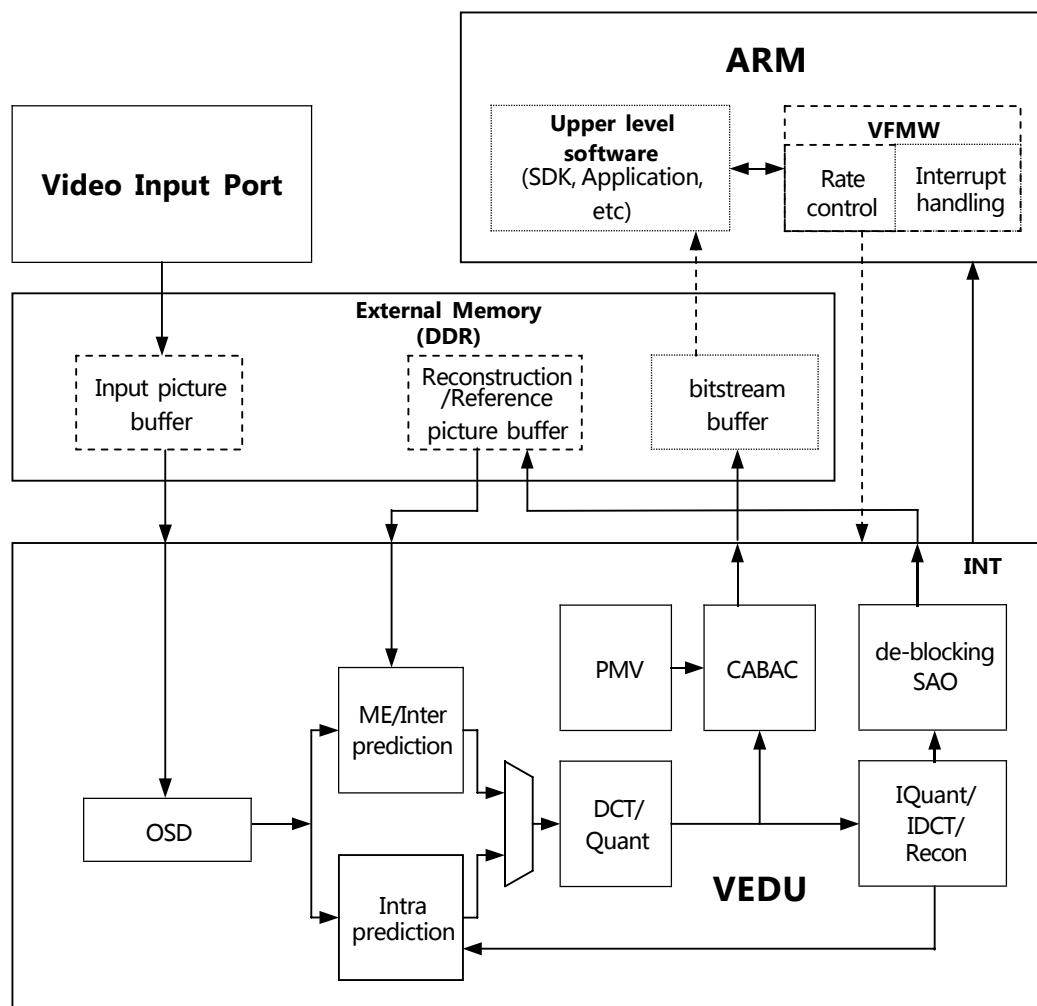
- Reconstruction/Reference image buffer

The VEDU writes reconstruction images to this buffer during encoding. These reconstruction images are used as the reference images of subsequent images. During the encoding of P frames, reference images are read from this buffer.

- Stream buffer

This buffer stores encoded streams. The VEDU writes streams to this buffer during encoding. This buffer is read by software.

**Figure 6-1** Encoding functional block diagram of the VEDU





## 6.3 JPGE

### 6.3.1 Overview

The JPGE provides high-performance JPEG encoding performance by using hardware. It supports snapshot and HD MJPEG encoding services.

### 6.3.2 Features

The JPGE has the following features:

- Supports ISO/IEC 10918-1(CCITT T.81) Baseline Process (DCT Sequential) encoding.
  - Supports Semi-Planar YUV420 and Semi-Planar YUV422 image encoding.
  - Supports configuration of quantization tables. An independent quantization table is provided for the Y component, U component, and V component, respectively.
  - Supports the color-to-gray function.
  - Supports the low-delay input function.
  - Supports the low-delay wrapping function.
  - Supports input cropping.
  - Supports input image formats: Semi-Planar YUV420 and Semi-Planar YUV422.
  - Supports the following encoding performance: 3200 x 1800 @ 30 fps encoding
  - Supports configurable image resolutions.
    - Minimum image resolution: 32 x 32
    - Maximum image resolution: 4096 x 4096
    - Supports the image width or height step of 2.
  - Supports OSD front-end overlaying.
    - OSD overlaying before encoding for a maximum of eight regions
    - OSD overlaying with any size and at any position (within the size and position ranges of the image)
    - 256-level alpha blending
    - OSD overlaying control (enabled or disabled)
    - OSD ARGB1555, ARGB4444, CLUT2, and CLUT4 formats
  - Supports the MJPEG output bit rate ranging from 2 kbit/s to 60 Mbit/s.

### 6.3.3 Function Description

[Figure 6-2](#) shows the functional block diagram of the JPGE. Based on the protocols that require a large number of operands, the JPGE supports OSD, level shift, discrete cosine transform (DCT), quantization, scanning, VLC encoding, and stream

generation. The ARM software configures quantization tables and handles interrupts.

Before the JPGE is enabled for video encoding, the software allocates two types of buffers for the JPGE in the external DDR SDRAM:

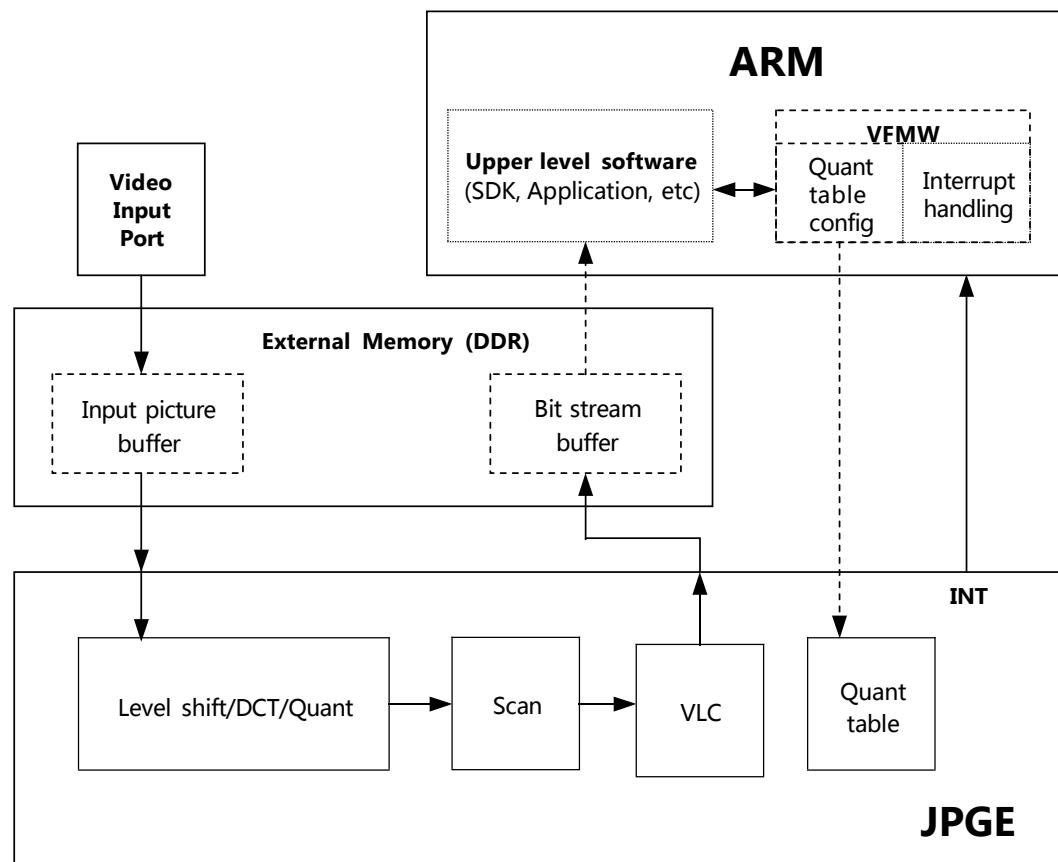
- Input image buffer

The JPGE reads the source images to be encoded from this buffer during encoding. This buffer is written by the VICAP module.

- Stream buffer

This buffer stores encoded streams. The JPGE writes streams to this buffer during encoding. This buffer is read by software.

**Figure 6-2** Encoding functional block diagram of the JPGE





## Contents

<b>7 Video and Graphics Processing .....</b>	<b>7-1</b>
<b>7.1 VPSS.....</b>	<b>7-1</b>
<b>7.1.1 Overview.....</b>	<b>7-1</b>
<b>7.1.2 Features .....</b>	<b>7-2</b>
<b>7.2 VGS .....</b>	<b>7-2</b>
<b>7.2.1 Overview.....</b>	<b>7-2</b>



# 7

## Video and Graphics Processing

### 7.1 VPSS

#### 7.1.1 Overview

The video processing subsystem (VPSS) implements video processing. It supports two operating modes—online (VICAP-VIPROC-VPSS all online) and offline (VPSS offline or VIPROC-VPSS online), and supports the functions such as video covering, cropping, scaling, single-component luminance processing , compression, mirroring, flipping, and cornered rectangles.

The VPSS has the following features:

- Video source up to 3840 pixels wide
- A maximum of three video output channels.
- Output of eight video rectangle regions covered by solid quadrilaterals.
- Video cropping and mirroring/flipping functions of three output channels
- Video data compression
- Input data in Semi-Planar YUV420/Semi-Planar YUV422/single component format
- Input data in linear format
- Output data in Semi-Planar YUV420/Semi-Planar YUV422/single component format
- Output data in linear or tile format
- Single-component luminance processing
- Low-delay mode for output channels
- Low-delay wrapping function for output channels
- Cornered rectangles for output channels



## 7.1.2 Features

- Scaling: Low-frequency filtering is supported when the input and output resolutions are different.
  - In offline mode, channel 0 supports 1/15x–16x horizontal or vertical scaling, and channels 1–2 support 1x–1/15x downscaling.
  - In online mode, channel 0 supports 1x–1/15x downscaling, and channels 1–2 support 1x–15x downscaling.
- Low-delay wrapping: By means of interaction with the encoder, the buffer is saved.

## 7.2 VGS

### 7.2.1 Overview

The video graphics system (VGS) implements video and graphics processing. The functions include scaling, LDC, rotation, video cropping, video covering, and line drawing.

The VGS has the following features:

- Processing of the input video source up to 3840 pixels wide
- One video output channel
- Supported video input formats: Semi-Planar YUV420/Semi-Planar YUV422
- Supported video output formats: Semi-Planar YUV420/Semi-Planar YUV422/package422
- One video region covered by a solid quadrilateral
- Video cropping
- Single-component processing
- Line drawing
- Rotation by 90° and 270°
- Scaling: Low-frequency filtering is supported when the input and output resolutions are different. The scaling ratio ranges from 1/30x to 16x.
- Mirror/flip function for output channels



## Contents

<b>8 Intelligent Video Engine .....</b>	<b>8-1</b>
8.1 NPU Acceleration Engine.....	8-1
8.1.1 Overview.....	8-1
8.1.2 Features .....	8-2
8.2 IVE .....	8-2
8.2.1 Overview.....	8-2
8.2.2 Features .....	8-3



## Figures

---

**Figure 8-1** Position of the NPU in the system.....8-1

**Figure 8-2** Position of the IVE in the system ..... 8-3

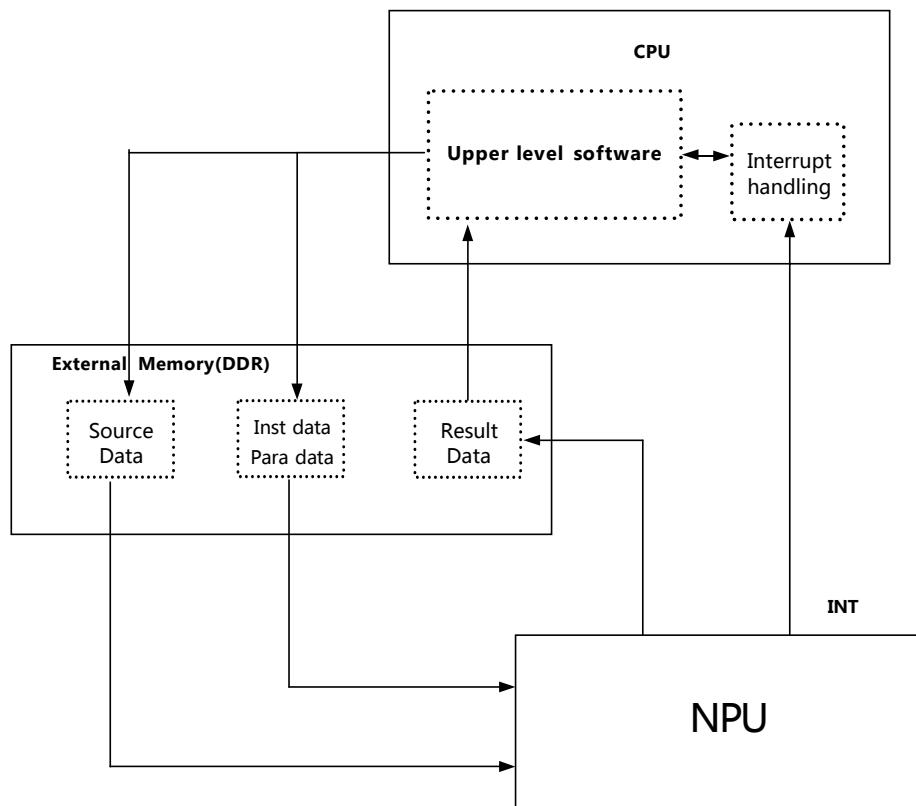
# 8 Intelligent Video Engine

## 8.1 NPU Acceleration Engine

### 8.1.1 Overview

The neural network processing unit (NPU) is a deep learning accelerator based on neural network structures such as convolutional neural network (CNN) and region-based CNN (RCNN). It can be used in application scenarios such as image classification and object detection. [Figure 8-1](#) shows the position of the NPU acceleration engine in the system.

**Figure 8-1** Position of the NPU in the system





## 8.1.2 Features

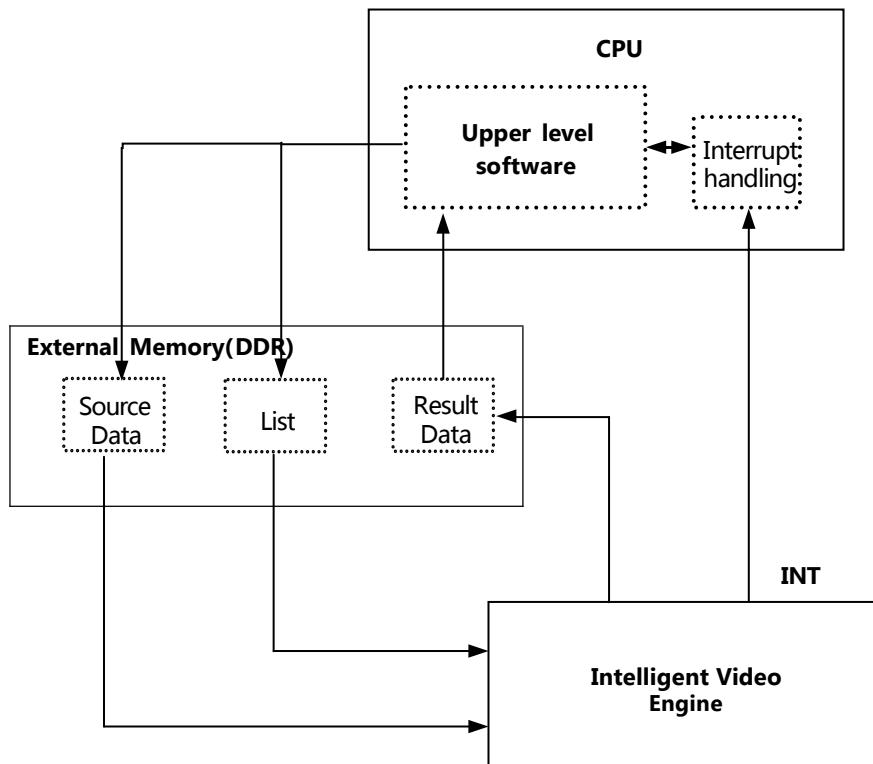
The NPU acceleration engine has the following features:

- Supports TOPS compute power
  - Hi3516CV610-10B: 0.5 TOPS
  - Hi3516CV610-20B/20S/20G/00B/00S/00G: 1 TOPS
- Supports FP16 vector operations.
- Supports the 4-bit/8-bit/16-bit data and parameter modes.
- Supports 4-bit/8-bit parameters.
- Supports the single-channel (grayscale) and tri-channel (RGB) of input images.
- Supports image preprocessing.
- Supports image batch processing.
- Supports the result reporting for the middle layer.

## 8.2 IVE

### 8.2.1 Overview

The intelligent video engine (IVE) module is used to accelerate hardware processing. It provides a series of basic calculation functions and some time-consuming functions used in the intelligent analysis algorithm.

**Figure 8-2** Position of the IVE in the system

## 8.2.2 Features

The IVE has the following features:

- DMA: Supports direct copying, alternative copying, memory filling, and mask copying.
- Filter: Supports 5x5 template filtering.
- CSC: Supports the color space conversion of YUV2RGB and RGB2YUV.
- Sobel: Supports the Sobel-like gradient calculation of 5x5 template.
- MagAndAng/Canny: Supports 5x5 template calculation and Canny edge extraction.
- Erode: Supports 5x5 template erosion.
- Dilate: Supports 5x5 template dilation.
- Threshold/Threshold\_s16/Threshold\_u16: Supports image threshold processing.
- And/Or/Xor: Supports the AND, OR, or XOR operation on two images.
- Add/Sub: Deals with the weighted plus and minus of two images.
- Integ: Supports the integral calculation.
- Hist: Supports the histogram statistics.



- Map: Assigns values to images by using 256-level mapping.
- 16BitTo8Bit: Performs linear conversion from 16-bit data to 8-bit data.
- OrdStatFilter: Filters sequential statistics (median value, maximum value, and minimum value).
- NCC: Calculates the correlation coefficients of two images with the same size.
- CCL: Marks connected regions.
- LBP: Supports the calculation in simple partial thresh mode.
- NormGrad: Performs a normalized gradient calculation.
- LKOpticalFlowPyr: Tracks the LK optical flow.
- STCorner: Detects ShiTomasi corner points.
- SAD: Calculates the accumulated sum of the absolute pixel differences corresponding to two images by block.
- Resize: Supports bilinear resizing.
- GMM2: Creates the GMM quickly for the grayscale image and RGB image.
- PSP: Supports perspective transformation.
- ACC: Supports image accumulation calculation.
- ImageDot: Calculates the dot product of images.
- AplhaBlending: Supports alpha blending calculation of images.

---

**NOTICE**

For details about the operator specifications, see *IVEAPIReference*.

---



# Contents

<b>9 Video Interfaces.....</b>	<b>9-1</b>
9.1 VI.....	9-1
9.1.1 Overview.....	9-1
9.1.2 Features .....	9-2
9.1.3 Function Description .....	9-3
9.1.4 Operating Mode .....	9-11
9.1.5 Register Summary.....	9-15
9.1.6 Register Description .....	9-19
9.2 MIPI Rx .....	9-58
9.2.1 Overview.....	9-58
9.2.2 Features .....	9-58
9.2.3 Function Description .....	9-59
9.2.4 Operating Mode of the MIPI RX Controller.....	9-69
9.2.5 MIPI RX Register Summary.....	9-70
9.2.6 MIPI RX Register Description .....	9-75



## Figures

<b>Figure 9-1</b> Functional block diagram of the VI module .....	9-1
<b>Figure 9-2</b> Vertical timing of the 720-line video system.....	9-5
<b>Figure 9-3</b> ITU-R BT.601 horizontal timing .....	9-5
<b>Figure 9-4</b> VS timings in the NTSC standard.....	9-6
<b>Figure 9-5</b> VS timings in the PAL standard .....	9-7
<b>Figure 9-6</b> DC horizontal timing .....	9-7
<b>Figure 9-7</b> DC vertical pulse timing .....	9-8
<b>Figure 9-8</b> DC vertical line active timing.....	9-8
<b>Figure 9-9</b> Relationship between the active video area and the horizontal/vertical blanking regions .....	9-9
<b>Figure 9-10</b> YUV4:2:2 storage mode.....	9-10
<b>Figure 9-11</b> RAW DATA storage mode .....	9-11
<b>Figure 9-12</b> Little-endian storage mode .....	9-11
<b>Figure 9-13</b> VICAP hardware workflow .....	9-12
<b>Figure 9-14</b> Software Configuration.....	9-13
<b>Figure 9-15</b> VIPROC hardware workflow in offline mode.....	9-14
<b>Figure 9-16</b> VIPROC software configuration workflow in offline mode .....	9-14
<b>Figure 9-17</b> MIPI RX workflow and its position in the system.....	9-58
<b>Figure 9-18</b> CSI-2 data packet transmission mechanism.....	9-61
<b>Figure 9-19</b> CSI-2 long packet format .....	9-61
<b>Figure 9-20</b> CSI-2 short packet format.....	9-62
<b>Figure 9-21</b> Image data format of the MIPI interface .....	9-62
<b>Figure 9-22</b> LVDS synchronization code and image transfer mode .....	9-65
<b>Figure 9-23</b> LVDS timing of a single pixel.....	9-65
<b>Figure 9-24</b> LVDS synchronization mode 1.....	9-66



**Figure 9-25** LVDS synchronization mode 2.....9-67

**Figure 9-26** Data format in HiSPi Packetized-SP mode .....9-68



## Tables

<b>Table 9-1</b> Format of the ITU-R BT.656 YUV4:2:2 line data .....	9-4
<b>Table 9-2</b> Formats of the SAV and EAV .....	9-4
<b>Table 9-3</b> Variables in the offset addresses of registers .....	9-15
<b>Table 9-4</b> Summary of VICAP registers (base address: 0x1740_0000) .....	9-15
<b>Table 9-5</b> Interfaces supported by the MIPI RX.....	9-59
<b>Table 9-6</b> LVDS synchronization code format.....	9-63
<b>Table 9-7</b> Sample of the fourth field of the LVDS synchronization code .....	9-64
<b>Table 9-8</b> HiSPi transfer modes.....	9-68
<b>Table 9-9</b> Variables in the MIPI RX register offset addresses.....	9-70
<b>Table 9-10</b> Summary of MIPI RX registers (base address: 0x173C_0000) .....	9-71



# 9 Video Interfaces

## 9.1 VI

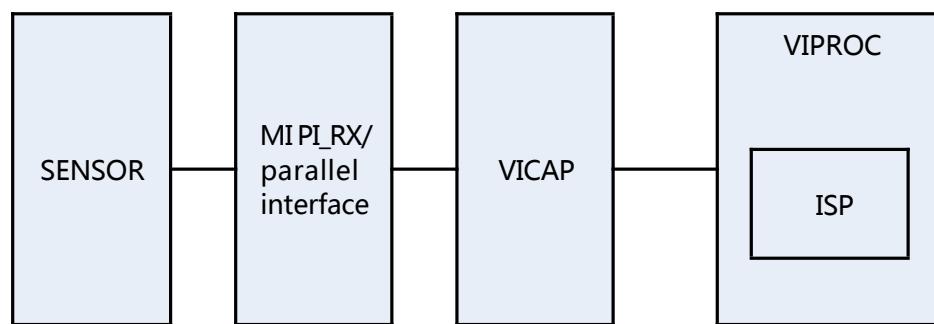
### 9.1.1 Overview

The video input (VI) receives video data over the MIPI RX (including MIPI, LVDS, and HiSPi) interfaces, BT.656/601 interfaces, and digital camera (DC) and stores the received data into the specified memory area. With an embedded ISP image processing unit, the VI module can directly interconnect with external raw data (Bayer RGB data). [Figure 9-1](#) shows the functional block diagram of the VI module.

The VI module consists of two physical submodules: the capture submodule VICAP and processing submodule VIPROC.

- The VICAP captures data of a single video input and stores the captured data in the DDR SDRAM or transmits the data to the VIPROC online. VIPROC processes video data in offline mode (reading data from the DDR) or online mode (receiving data from VICAP online).
- The VIPROC allows the processed data to be written to the DDR SDRAM or directly transmits the processed data to the VPSS online.

**Figure 9-1** Functional block diagram of the VI module





## 9.1.2 Features

The VICAP submodule has the following features:

### NOTE

For details about the VICAP working clock, see the description in section 3.2.5 "CRG Register Description."

- A maximum input width of 3840 and maximum resolution of 3840 x 2160.
  - 2880 x 1620 for Hi3516CV610-10B
- A maximum of two sensor data inputs
  - Interface 0 supports a maximum of 2F-WDR raw inputs of 3840 x 2160@20 fps or 3200 x 1800@30 fps.
  - Interface 1 supports a maximum of raw inputs of 3840 x 2160@20 fps or 3200 x 1800@30 fps.
- A maximum of 14-bit data width for each VICAP
- Progressive input mode
- BT.656, BT.601, and DC interfaces
- MIPI, LVDS, and HiSPi interfaces
- YUV data input over the MIPI
  - Semi-planar YUV422/Semi-planar YUV420 formats
- Flash trigger
- Shutter trigger
- Sensor master mode and slave mode
- RAW data compression
- Image data output to VIPROC in online mode
- Image data output to the DDR SDRAM in offline mode
  - Two output channels
  - CH0 supports only RAW data output.
  - CH1 supports RAW and YUV data output.
- Output data formats in offline mode:
  - Semi-planar YUV422/Semi-planar YUV420/Single component
  - RAW mode

The VIPROC submodule has the following features:

- Data processing in online mode (data transmitted from the VICAP online)
- Data processing in offline mode (data transmitted from the DDR SDRAM)
- A maximum processing performance of 3840 x 2160@20 fps/3200 x 1800@30 fps (after WDR combination)
- Embedded ISP processing



- 1-channel video output
- Image data output to the DDR
- Output data formats:
  - Semi-planar YUV422/Semi-planar YUV420/Single component
  - RAW mode

## 9.1.3 Function Description

### 9.1.3.1 Typical Applications

The VI module is a video input capture processing unit that supports multiple timing inputs. Different functional modes can be configured so that the VI module can flexibly adapt to different input video interfaces.

The VI module can process 2-channel video input signals.

The VI module supports the following typical inputs:

- 1-channel [3840 x 2160@20 fps](#) input of 2F-WDR data
- 1-channel [3200 x 1800@30 fps](#) input of 2F-WDR data
- 2-channel 3840 x 2160@20 fps input of RAW data
- 2-channel 3200 x 1800@30 fps input of RAW data
- 1-channel 3200 x 1800@30fps input of RAW data, and 1-channel 2M (1920 x 1080)@30 fps input of YUV data

### 9.1.3.2 Function Implementation

#### BT.656 Interface Timings

The VICAP module supports the progressive YUV422 data input in the BT.656 timing format.

- Horizontal timing

Based on the ITU-R BT.656 protocol, sync signals are included in data streams. The special bytes start of active video (SAV) and end of active video (EAV) indicate the start and end of the active line data respectively. In video data streams, the header of the timing reference code indicates that the following byte is SAV or EAV. The timing reference code consists of FF 00 00. FF and 00 indicate the reserved bytes of the image encoding data, that is, non-image data. [Table 9-1](#) shows the format of the ITU-R BT.656 line data.



**Table 9-1** Format of the ITU-R BT.656 YUV4:2:2 line data

Timing Reference Code				Line Blanking Region					Timing Reference Code				YUV4:2:2 with 720 Active Pixels							
FF	00	00	EA	80	1	...	8	1	FF	0	00	SA	V	U0	Y0	V0	Y1	...	V718	Y719

The difference between the SAV and EAV depends on the special bit H. Both the SAV and EAV include the vertical blanking bit V and field indicator bit F. For details about the SAV and EAV, see [Table 9-2](#).

**Table 9-2** Formats of the SAV and EAV

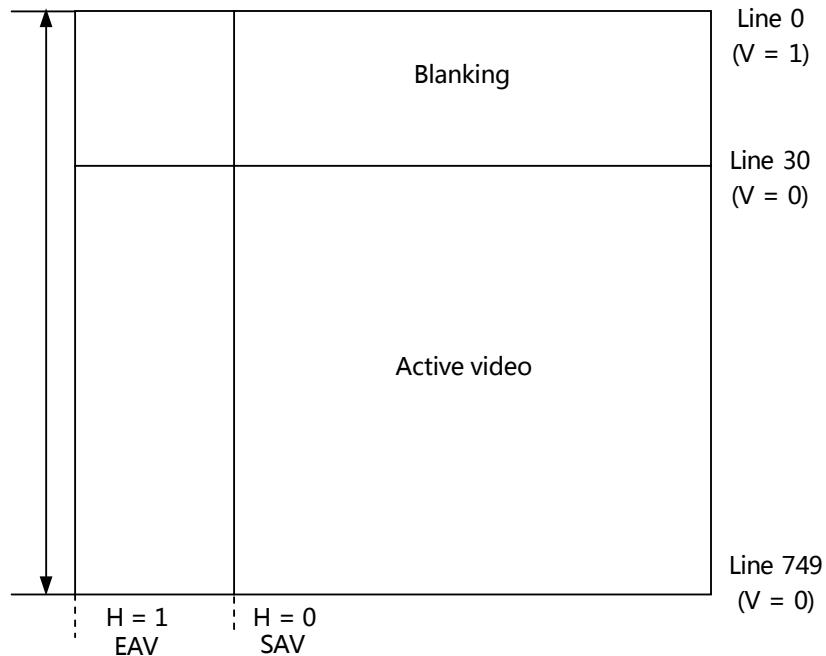
bit[7]	bit[6](F)	bit[5](V)	bit[4](H)	bit[3:0](P3-P0)
Fixed value 1	Field indicator bit First field: F = 0 Second field: F = 1	Vertical blanking bit VBI: V=1 Active video: V=0	SAV: H=0 EAV: H=1	Check bits

- Vertical timing

The positions of the vertical timings are determined by bit V of the timing reference codes SAV and EAV. [Figure 9-2](#) shows the vertical timing of the 720-line video system.



**Figure 9-2** Vertical timing of the 720-line video system



The VICAP module identifies vertical timings based on SAV and EAV regardless of the lines.

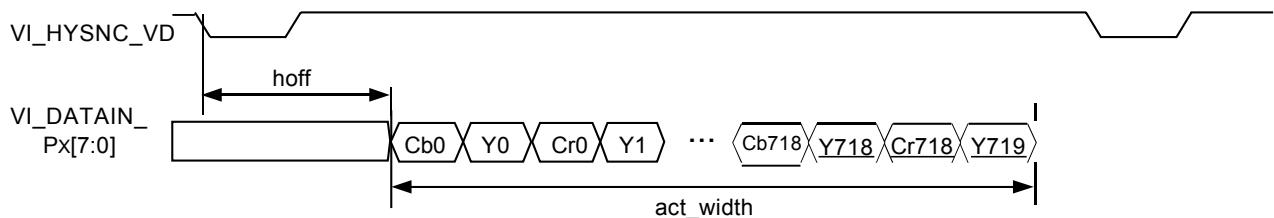
## BT.601 Interface Timings

The VICAP module supports the progressive YUV422 data input in the BT.601 timing format.

- Horizontal timing

The horizontal pulse indicates the start of a new line, as shown in [Figure 9-3](#). After hoff clock cycles, an input signal passes the line front blanking region and enters the line active data region. The value of hoff is 244 in the NTSC 525-line system or 264 in the PAL 625-line system. After act\_width clock cycles, the input signal passes the line active data region and enters the line back blanking region. The value of act\_width can be configured, and the typical value is 720 or 704. In addition, the horizontal sync (HS) polarity is configurable.

**Figure 9-3** ITU-R BT.601 horizontal timing



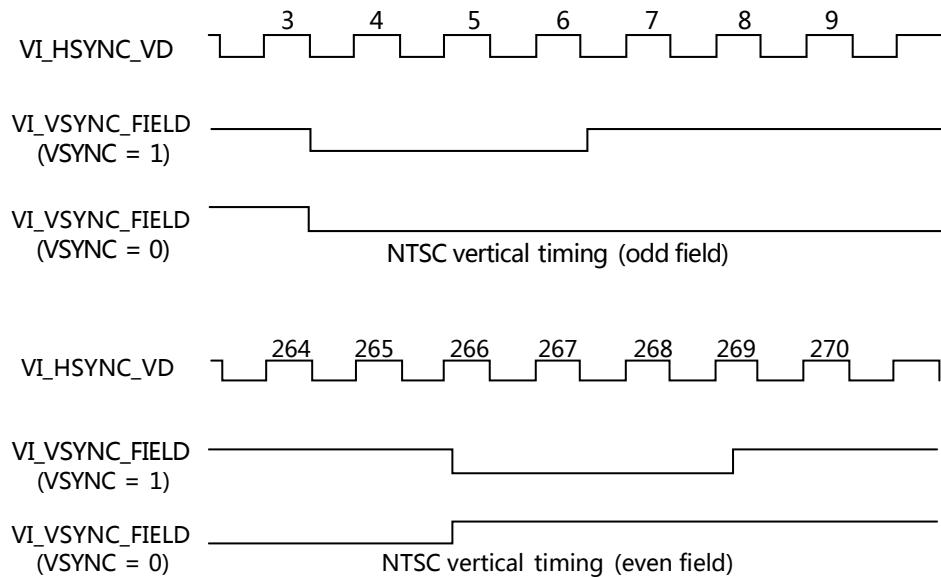


- Vertical timings

Based on the ITU-R BT.601 recommendation, the signal VSYNC is the vertical sync (VS) signal and indicates the start of the VSYNC pulse frame. The VICAP module supports two vertical synchronization methods.

[Figure 9-4](#) shows VI VS timings in the NTSC standard (525 lines), and [Figure 9-5](#) shows VI VS timings in the PAL standard (625 lines). VI\_HSYNC\_VD indicates the HS pulse, and VI\_VSYNC\_FIELD indicates the VS pulse when VSYNC is 1 or field sync signal when VSYNC is 0.

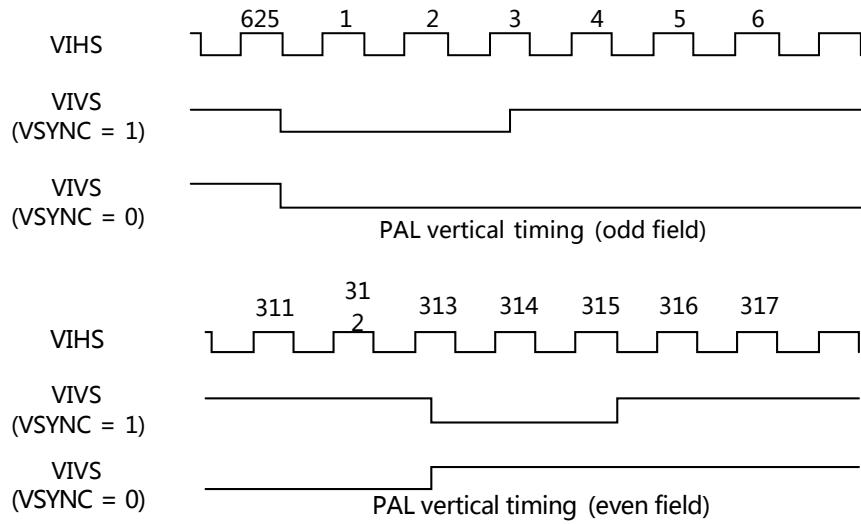
**Figure 9-4** VS timings in the NTSC standard



In NTSC interlaced scanning mode, the level of the VS signal in field 1 becomes low in the start position of line 4, retains low for three consecutive lines, and then becomes high in the start position of line 7. The VICAP module receives 240-line data from line 22 to line 261. The level of the VS signal in field 2 becomes low in the middle of line 266, retains low for three consecutive lines, and then becomes high in the middle of line 269. The VICAP module receives 240-line data from line 285 to line 524.



**Figure 9-5 VS timings in the PAL standard**



In PAL interlaced scanning mode, the level of the VS signal in field 1 becomes low in the start position of line 1, retains low for 2.5 consecutive lines, and then becomes high in the middle of line 3. The VICAP module receives 288-line data from line 24 to line 310. The level of the VS signal in field 2 becomes low in the middle of line 313, retains low for 2.5 consecutive lines, and then becomes high in the start position of line 316. The VICAP module receives 288-line data from line 336 to line 623.

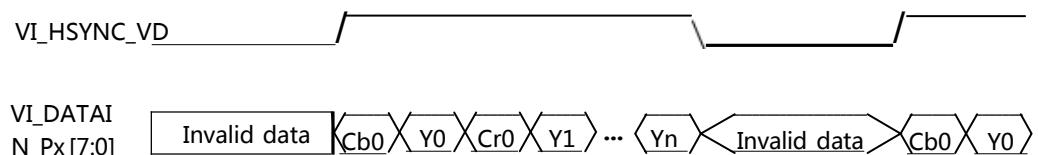
The preceding timings are typical BT.601 vertical timings. The number of lines from the start of the field to the active line, the number of field active lines, and VS polarity can be configured.

## DC Interface Timings

- Horizontal timing

When a DC is connected to the VICAP module, VI\_HSYNC\_VD is the data valid signal. The polarity of this signal is configurable. [Figure 9-6](#) shows the DC horizontal timing.

**Figure 9-6 DC horizontal timing**

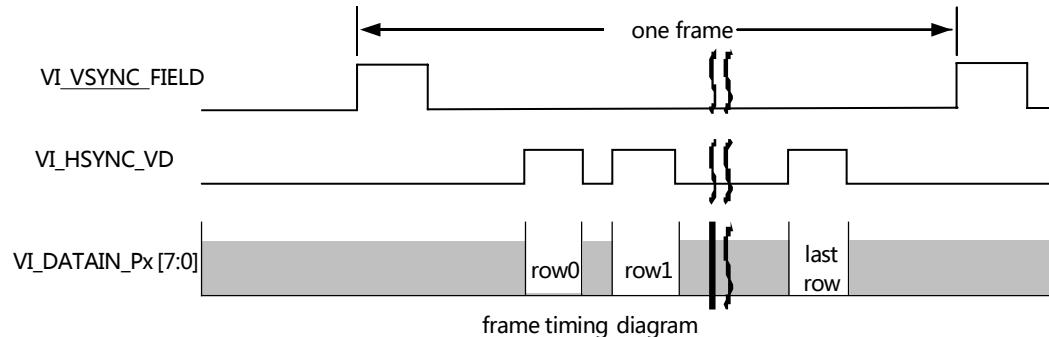


- Vertical timing

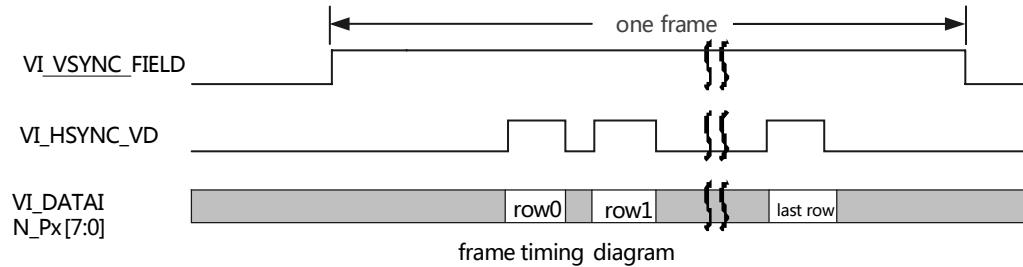


The VICAP module supports the vertical pulse timing and vertical line active timing, as shown in [Figure 9-7](#) and [Figure 9-8](#). The VS polarity is configurable.

**Figure 9-7 DC vertical pulse timing**



**Figure 9-8 DC vertical line active timing**

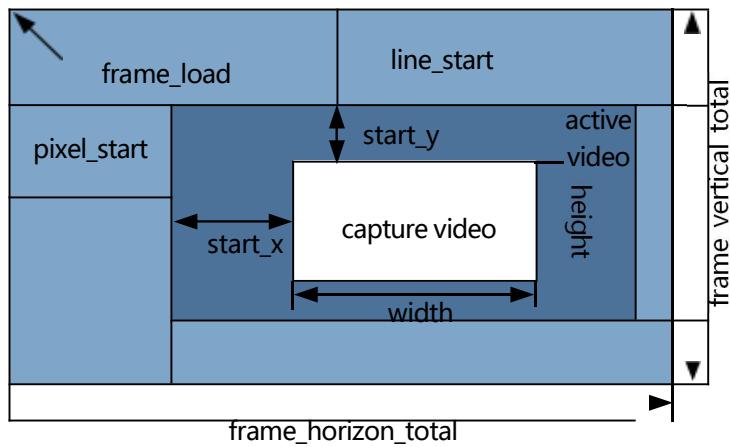


The preceding two timings are the same from the aspect of internal processing of the VICAP module. To be specific, the VICAP module considers the start of a frame after it detects a rising edge or a falling edge, and then detects the data active signal to check whether the current data is active.

### 9.1.3.3 Image Cropping

An active video starts from the end of the horizontal blanking region and vertical blanking region. The actual view range, however, is within the active video range. That is, compared with the boundary of the active video. See [Figure 9-9](#).

**Figure 9-9** Relationship between the active video area and the horizontal/vertical blanking regions



#### 9.1.3.4 Image Storage Modes

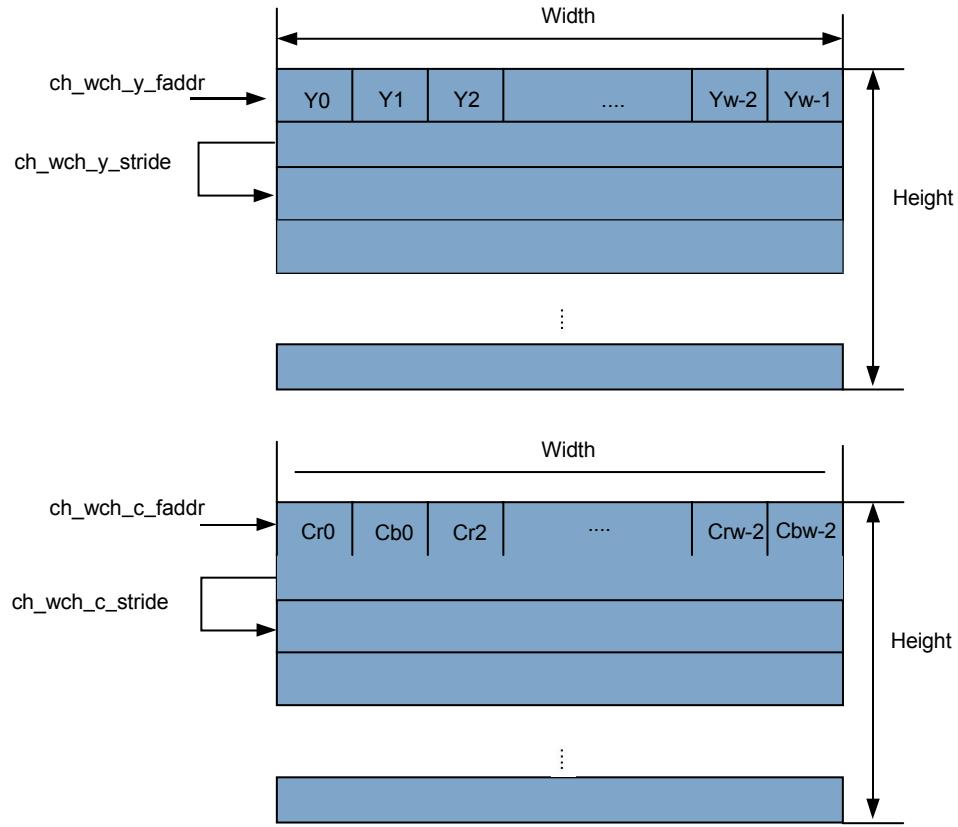
The supported image storage modes include:

1. YUV data storage
  - Semi-planar YUV storage mode  
After setting a view area, the system stores the read data in semi-planar mode. That is, the luminance component and the chrominance component are stored in the luminance space and chrominance space of the DDR respectively.
  - For one line, the luminance component and chrominance component are stored separately and consecutively.
  - For two consecutive lines, data is stored based on the stride parameter that defines the storage stride between the beginnings of two lines.
  - The storage positions of the luminance component and chrominance component in the DDR are specified by the start address **base\_addr**.

[Figure 9-10](#) shows the mode of storing the YUV4:2:2 data captured by the VICAP module.



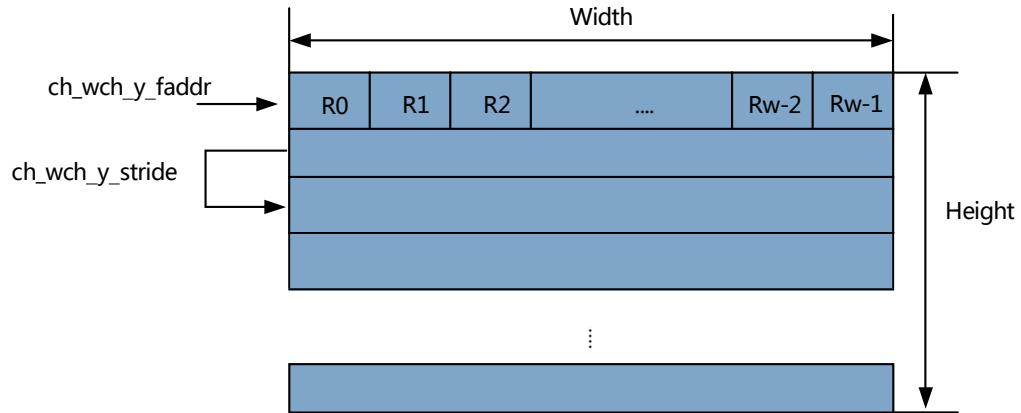
Figure 9-10 YUV4:2:2 storage mode



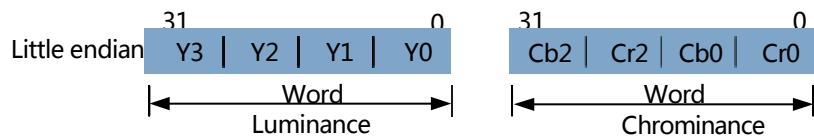
## 2. Raw data storage

- The raw data is stored in single-component mode.
- The raw data is consecutively stored in one line.
- For two consecutive lines, data is stored based on the stride parameter that defines the storage stride between the beginnings of two lines.
- The storage position of the DDR is specified by the start address **base\_addr**.

[Figure 9-11](#) shows the mode of storing the raw data captured by the VICAP module.

**Figure 9-11 RAW DATA storage mode**

In the DDR, data is stored by word (32 bits). Four 8-bit pixels constitute a 32-bit word. See [Figure 9-12](#).

**Figure 9-12 Little-endian storage mode**

The VICAP module supports the DDR that stores data in little-endian mode. The storage address is 8-byte aligned.

## 9.1.4 Operating Mode

### 9.1.4.1 reg\_newer Function

- Before enabling the channel of the VICAP module, the software must perform the following operations:
  - Configures the VICAP attribute register.
  - Writes 1 to the reg\_newer bit to inform the VICAP module that the current register is ready.
- After the VICAP module is enabled, the VICAP logic starts to work. When a frame arrives:
  - If reg\_newer is 0, the VICAP module does not receive data. However, it sets the hardware status to snooze (the following hardware statuses are shown in [Figure 9-13](#)) and waits for the next frame.
  - If reg\_newer is 1, the VICAP module starts receiving data, generates the register update interrupt (reg\_update\_int), and sets the hardware to busy status.

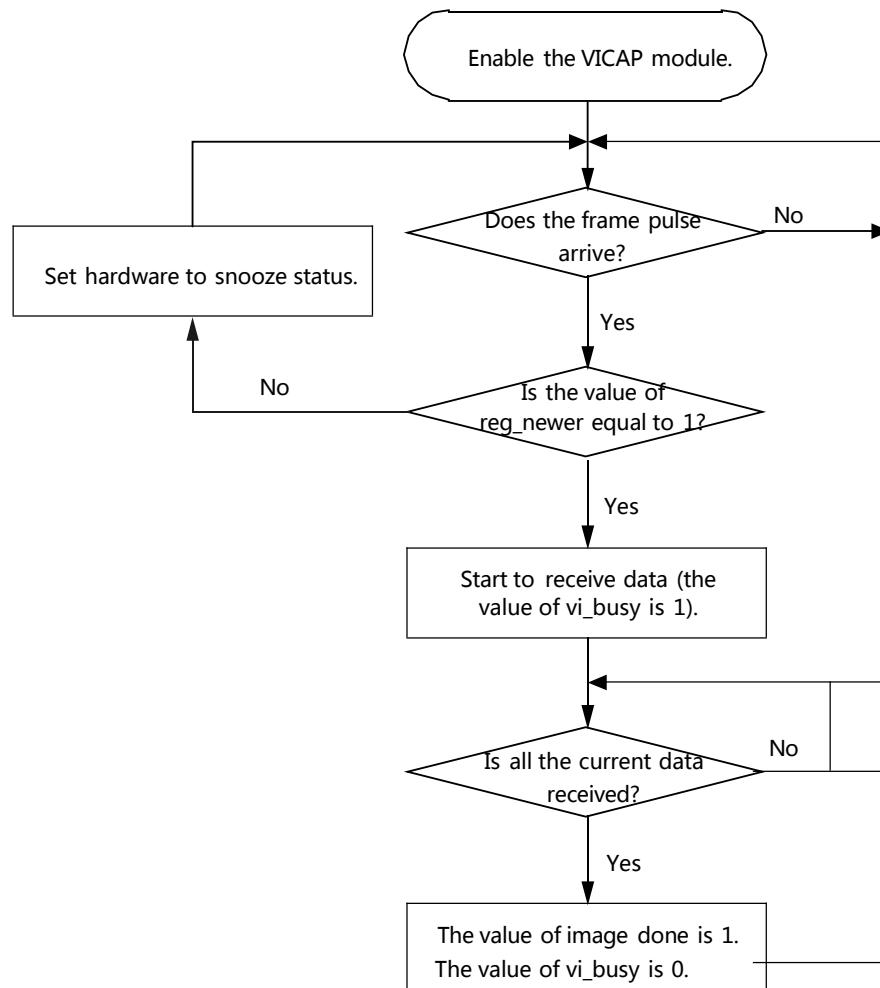


- After all the current data is received, the busy status of the hardware is cleared. When the next frame arrives:
  - If reg\_newer is 0, the VICAP module does not receive the data of the next frame.
  - If reg\_newer is 1, the VICAP module receives the data of the next frame.

### 9.1.4.2 VICAP Hardware Workflow in Online Mode

Figure 9-13 shows the VICAP hardware workflow.

Figure 9-13 VICAP hardware workflow



Each time after the VICAP module receives the data of a frame, it checks the reg\_newer bit when the next frame arrives. If reg\_newer is 1, software has updated or checked corresponding VICAP registers. In this case, the VICAP module automatically loads the register values configured by the software to the working register (this register is inaccessible to software), clears reg\_newer, and starts to

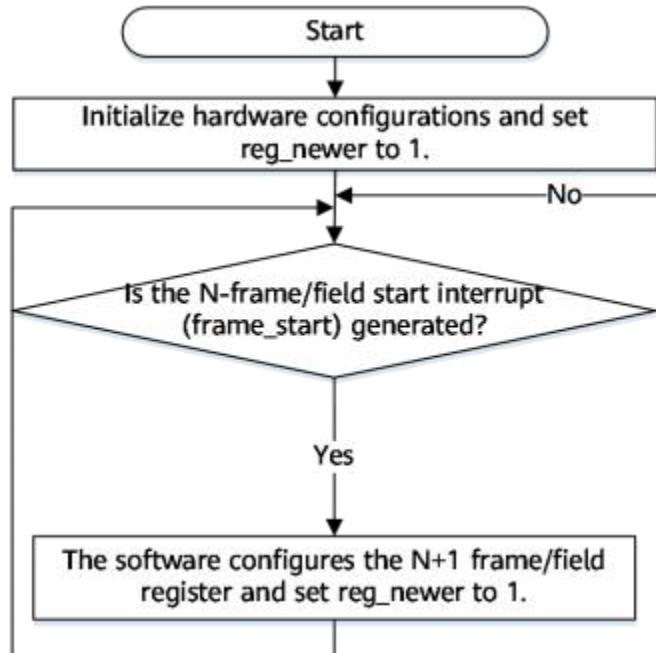


receive the data of the next frame. If reg\_newer is 0, the VICAP module starts to receive data only when reg\_newer is 1 and a new frame arrives.

### 9.1.4.3 Software Configuration Workflow in Online Mode

Figure 9-14 shows the software configuration process in interrupt mode.

**Figure 9-14 Software Configuration**

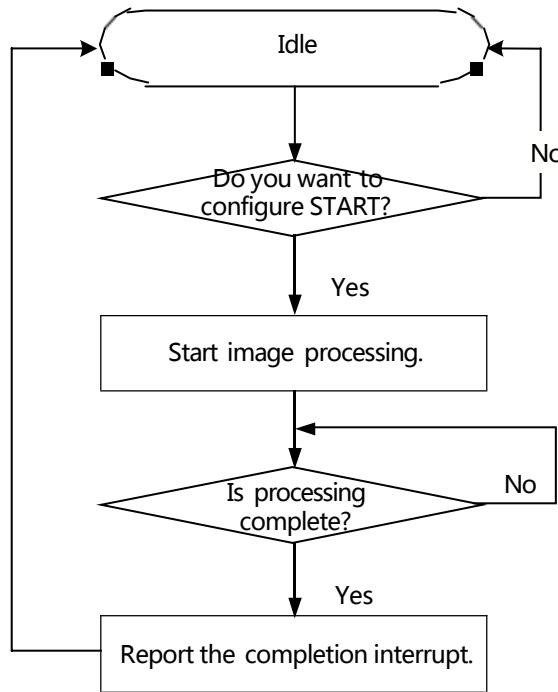


### 9.1.4.4 VIPROC Hardware Workflow in Offline Mode

The VIPROC submodule supports the offline mode. It can process multi-channel data in time-division multiplexing mode, as shown in Figure 9-15.



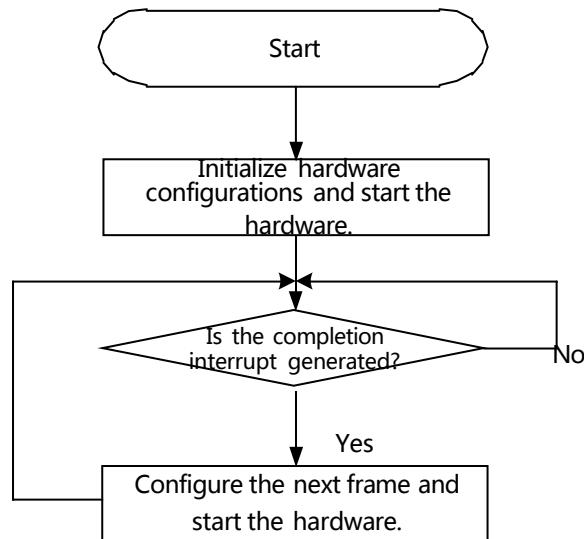
**Figure 9-15** VIPROC hardware workflow in offline mode



#### 9.1.4.5 VIPROC Software Configuration Workflow in Offline Mode

Figure 9-16 shows the VIPROC software configuration workflow.

**Figure 9-16** VIPROC software configuration workflow in offline mode





## 9.1.5 Register Summary

[Table 9-3](#)describes the value ranges and meanings of the variables in the offset addresses of registers.

**Table 9-3** Variables in the offset addresses of registers

Variable Name	Value Range	Description
CH_N	0-1	Two output channels of the VICAP module
PT_N	0-1	Two input channels of the VICAP module

[Table 9-4](#)describes the VICAP registers.

**Table 9-4** Summary of VICAP registers (base address: 0x1740\_0000)

Offset address	Name	Description	Page
0x0000	WK_MODE	Global operating mode configuration register	<a href="#">9-19</a>
0x0010	AXI_CFG	Bus configuration register	<a href="#">9-19</a>
0x0054	CMUX_ID_CFG	CMUX frame ID select register	<a href="#">9-20</a>
0x0058	ISP_MODE	ISP mode select register	<a href="#">9-20</a>
0x00E0	VICAP_INT1	Interrupt indicator register	<a href="#">9-21</a>
0x00E4	VICAP_INT1_STATE	Interrupt status register	<a href="#">9-22</a>
0x00E8	VICAP_INT1_MASK	Interrupt mask configuration register	<a href="#">9-23</a>
0x00F0	VICAP_INT0	Interrupt indicator register	<a href="#">9-25</a>
0x00F4	VICAP_INT0_STATE	Interrupt status register	<a href="#">9-26</a>
0x00F8	VICAP_INT0_MASK	Interrupt mask configuration register.	<a href="#">9-27</a>
0x0200	ONLINE0_CFG	Online channel 0 control register	<a href="#">9-28</a>
0x02AC	ONLINE0_SIZE	Online 0 output active area size register	<a href="#">9-28</a>
0x030c	VICAP_AFIFO0_SIZE	VICAP_AFIFO size configuration register	<a href="#">9-29</a>



Offset address	Name	Description	Page
0x0fe4	VICAP_AXIO_WR_ERRO_R	AXI 0 write error status register	<a href="#">9-29</a>
0x0ff0	VICAP_DBG_INT	System abnormal interrupt status register	<a href="#">9-30</a>
0x0ff8	VICAP_DBG_INT_MAS_K	System abnormal mask status register	<a href="#">9-30</a>
0x1000+PT_N×0x100	PT_INTF_MOD	Interface mode register	<a href="#">9-31</a>
0x1010+PT_N×0x100	PT_OFFSET0	Component 0 offset register	<a href="#">9-31</a>
0x1014+PT_N×0x100	PT_OFFSET1	component 1 offset register	<a href="#">9-32</a>
0x1018+PT_N×0x100	PT_OFFSET2	component 2 offset register	<a href="#">9-32</a>
0x1020+PT_N×0x100	PT_BT656_CFG	BT.656 configuration register	<a href="#">9-33</a>
0x1038+PT_N×0x100	PT_UNIFY_TIMING_CFG	Timing configuration register	<a href="#">9-33</a>
0x103C+PT_N×0x100	PT_GEN_TIMING_CFG	Timing recovery module configuration register	<a href="#">9-35</a>
0x1040+PT_N×0x100	PT_UNIFY_DATA_CFG	Data configuration register	<a href="#">9-36</a>
0x1044+PT_N×0x100	PT_GEN_DATA_CFG	Data generation module configuration register	<a href="#">9-37</a>
0x1048+PT_N×0x100	PT_GEN_DATA_COEF	Data generation module coefficient register	<a href="#">9-38</a>
0x104C+PT_N×0x100	PT_GEN_DATA_INIT	Data generation initial value configuration register	<a href="#">9-39</a>
0x1050+PT_N×0x100	PT_YUV444_CFG	YUV444 configuration register	<a href="#">9-39</a>
0x1060+PT_N×0x100	PT_FSTART_DLY	Port fstart interrupt delay register	<a href="#">9-39</a>
0x1064+PT_N×0x100	PT_FSTART_H_DLY	Port fstart interrupt horizontal delay register	<a href="#">9-39</a>
0x1080+PT_N×0x100	PT_INTF_HFB	Horizontal front blanking width register	<a href="#">9-40</a>
0x1084+PT_N×0x100	PT_INTF_HACT	Horizontal active area width register	<a href="#">9-40</a>
0x1088+PT_N×0x100	PT_INTF_HBB	Horizontal back blanking width register.	<a href="#">9-40</a>



Offset address	Name	Description	Page
0x108C+PT_N×0x100	PT_INTF_VFB	Vertical front blanking width register	<a href="#">9-41</a>
0x1090+PT_N×0x100	PT_INTF_VACT	Vertical active area width register	<a href="#">9-41</a>
0x1094+PT_N×0x100	PT_INTF_VBB	Vertical back blanking width register	<a href="#">9-41</a>
0x1098+PT_N×0x100	PT_INTF_VBFB	Vertical bottom front blanking width register	<a href="#">9-42</a>
0x109C+PT_N×0x100	PT_INTF_VBACT	Vertical bottom active area width register	<a href="#">9-42</a>
0x10A0+PT_N×0x100	PT_INTF_VBBB	Vertical bottom back blanking width register	<a href="#">9-42</a>
0x10A4+PT_N×0x100	PT_ID_CFG	ID configuration register	<a href="#">9-42</a>
0x10E0+PT_N×0x100	PT_STATUS	Port status register	<a href="#">9-43</a>
0x10E4+PT_N×0x100	PT_BT656_STATUS	BT.656 status register	<a href="#">9-44</a>
0x10EC+PT_N×0x100	PT_SIZE	Input image active area size indicator register	<a href="#">9-44</a>
0x10F0+PT_N×0x100	PT_INT	Port interrupt indicator register	<a href="#">9-44</a>
0x10F8+PT_N×0x100	PT_INT_MASK	Port interrupt mask register	<a href="#">9-45</a>
0x10000+CH_N×0x1000	CH_CTRL	Channel control register	<a href="#">9-45</a>
0x10004+CH_N×0x1000	CH_REG_NEWER	Capture control register	<a href="#">9-46</a>
0x10034+CH_N×0x1000	CH_DLY_CFG	Channel input image start interrupt delay configuration register	<a href="#">9-46</a>
0x10038+CH_N×0x1000	CH_FRM_CNT	Latch frame ID register for a delay interrupt	<a href="#">9-47</a>
0x10040+CH_N×0x1000	CH_CROP_CFG	Channel crop enable register	<a href="#">9-47</a>
0x10044+CH_N×0x1000	CH_CROP0_START	Channel 0 region crop start register	<a href="#">9-47</a>
0x10048+CH_N×0x1000	CH_CROP0_SIZE	Channel 0 region crop size configuration register	<a href="#">9-48</a>
0x10050+CH_N×0x1000	CH_SKIP_Y_CFG	Channel Y component skip configuration register	<a href="#">9-48</a>



Offset address	Name	Description	Page
0x10058+CH_N×0x1000	CH_SKIP_C_CFG	Channel C component skip configuration register	<a href="#">9-48</a>
0x10080+CH_N×0x1000	CH_WCH_Y_CFG	Y component configuration register	<a href="#">9-48</a>
0x10084+CH_N×0x1000	CH_WCH_Y_SIZE	Y component storage size register for the WCH module	<a href="#">9-50</a>
0x10088+CH_N×0x1000	CH_WCH_Y_OFFSET	Y component offset configuration register	<a href="#">9-50</a>
0x10094+CH_N×0x1000	CH_WCH_Y_FADDR_L	Lower 32 bits register of the Y component storage base address in the WCH module	<a href="#">9-50</a>
0x100A0+CH_N×0x1000	CH_WCH_Y_STRIDE	Y component stride register in the WCH module	<a href="#">9-51</a>
0x100B0+CH_N×0x1000	CH_WCH_C_CFG	C component configuration register in the WCH module	<a href="#">9-51</a>
0x100B4+CH_N×0x1000	CH_WCH_C_SIZE	C component storage size register in the WCH module	<a href="#">9-52</a>
0x100C4+CH_N×0x1000	CH_WCH_C_FADDR_L	Lower 32 bits register of the C component storage base address in the WCH module	<a href="#">9-52</a>
0x100CC+CH_N×0x1000	CH_WCH_TUNL_HAD_DR_L	Lower 32-bit tunnel base address register	<a href="#">9-52</a>
0x100D0+CH_N×0x1000	CH_WCH_C_STRIDE	C component stride register in the WCH module	<a href="#">9-53</a>
0x100E8+CH_N×0x1000	CH_Y_ACT_SIZE	Output image active area size indicator register of the luminance channel	<a href="#">9-53</a>
0x100EC+CH_N×0x1000	CH_C_ACT_SIZE	Output image active area size indicator register of the chrominance channel	<a href="#">9-53</a>
0x100F0+CH_N×0x1000	CH_INT	Channel raw interrupt register	<a href="#">9-54</a>
0x100F8+CH_N×0x1000	CH_INT_MASK	Channel interrupt mask register	<a href="#">9-55</a>
0x10208+CH_N×0x1000	CH_WCH_STT_AE_HIS_T_ADDR_L	Lower 32 bits register of the ISP AE_HIST statistics storage base address	<a href="#">9-57</a>



Offset address	Name	Description	Page
0x10210+CH_N×0x1000	CH_WCH_STT_AE_AVE_R_R_GR_ADDR_L	Lower 32 bits register of the ISP AE_AVER_R_GR statistics storage base address	<a href="#">9-57</a>
0x10218+CH_N×0x1000	CH_WCH_STT_AE_AVE_R_GB_B_ADDR_L	Lower 32 bits register of the ISP AE_AVER_GB_B statistics storage base address	<a href="#">9-57</a>

## 9.1.6 Register Description

### WK\_MODE

WK\_MODE is the global operating mode configuration register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_0003

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	-	reserved	Reserved	0x1
[0]	RW	power_mode	Clock mode 0: The low-power mode is disabled. 1: The low-power mode is enabled.	0x1

### AXI\_CFG

AXI\_CFG is the bus configuration register.

Offset Address: 0x0010 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	Reserved	0x00000
[11:8]	RW	r_outstanding	Number of read request outstandings Value range: [0, 15]	0x0
[7:4]	RW	w_outstanding	Number of write request outstandings Value range: [0, 7]	0x0
[3:0]	-	reserved	Reserved	0x0



## CMUX\_ID\_CFG

CMUX\_ID\_CFG is the CMUX frame ID select register.

Offset Address: 0x0054 Total Reset Value: 0x0000\_00FF

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:4]	RW	chn1_id	Channel 1 ID select register 0: Data with this ID is filtered. 1: Data with this ID is received. bit[0]: Channel RX enable for data with the ID 0 bit[1]: Channel RX enable for data with the ID 1 bit[2]: Channel RX enable for data with the ID 2 bit[3]: Channel RX enable for data with the ID 3	0xF
[3:0]	RW	chn0_id	Channel 0 ID select register 0: Data with this ID is filtered. 1: Data with this ID is received. bit[0]: Channel RX enable for data with the ID 0 bit[1]: Channel RX enable for data with the ID 1 bit[2]: Channel RX enable for data with the ID 2 bit[3]: Channel RX enable for data with the ID 3	0xF

## ISP\_MODE

ISP\_MODE is the ISP mode select register.

Offset Address: 0x0058 Total Reset Value: 0x0300\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25]	RW	isp1_clk_en	ISP1 gating enable register	0x1



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[24]	RW	isp0_clk_en	ISP0 gating enable register 0: disabled 1: enabled	0x1
[23:18]	-	reserved	Reserved	0x00
[17]	RW	isp1_in_yuv_mode	ISP 1 mode select configuration register 0: Raw data is input. 1: YUV data is input.	0x0
[16]	RW	isp0_in_yuv_mode	ISP 0 mode select configuration register 0: Raw data is input. 1: YUV data is input.	0x0
[15:2]	-	reserved	Reserved	0x0000
[1]	RW	isp1_mode	ISP 1 mode select configuration register 0: Data is processed by the ISP. 1: Data is bypassed by the ISP.	0x0
[0]	RW	isp0_mode	ISP 0 mode select configuration register 0: Data is processed by the ISP. 1: Data is bypassed by the ISP.	0x0

## VICAP\_INT1

VICAP\_INT1 is an interrupt indicator register.

Offset Address: 0x00E0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26]	RO	int1_dbg	System bus exception interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[25]	-	reserved	Reserved	0x0
[24]	RO	int1_src0	SRC 0 interrupt indicator 0: No interrupt is generated.	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
			1: An interrupt is generated.	
[23:18]	-	reserved	Reserved	0x00
[17]	RO	int1_isp1	ISP 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[16]	RO	int1_isp0	ISP 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15:10]	-	reserved	Reserved	0x00
[9]	RO	int1_ch1	Channel 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	int1_ch0	Channel 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7:2]	-	reserved	Reserved	0x00
[1]	RO	int1_pt1	Port 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	int1_pt0	Port 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## VICAP\_INT1\_STATE

VICAP\_INT1\_STATE is an interrupt status register.

Offset Address: 0x00E4 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:27]	-	reserved	Reserved	0x00
[26]	RO	int1_state_dbg	System bus exception interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
[25]	-	reserved	Reserved	0x0
[24]	RO	int1_state_src0	SRC 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[23:18]	-	reserved	Reserved	0x00
[17]	RO	int1_state_isp1	ISP 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[16]	RO	int1_state_isp0	ISP 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15:10]	-	reserved	Reserved	0x00
[9]	RO	int1_state_ch1	Channel 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	int1_state_ch0	Channel 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7:2]	-	reserved	Reserved	0x00
[1]	RO	int1_state_pt1	Port 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	int1_state_pt0	Port 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## VICAP\_INT1\_MASK

VICAP\_INT1\_MASK is an interrupt mask configuration register.

Offset Address: 0x00E8 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26]	RW	mask1_dbg	System bus exception interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[25]	-	reserved	Reserved	0x0
[24]	RW	mask1_src0	SRC 0 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[23:18]	-	reserved	Reserved	0x00
[17]	RW	mask1_isp1	ISP 1 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[16]	RW	mask1_isp0	ISP 0 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[15:10]	-	reserved	Reserved	0x00
[9]	RW	mask1_ch1	Channel 1 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[8]	RW	mask1_ch0	Channel 0 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[7:2]	-	reserved	Reserved	0x00
[1]	RW	mask1_pt1	Port 1 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[0]	RW	mask1_pt0	Port 0 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0



## VICAP\_INT0

VICAP\_INT0 is the Interrupt indicator register.

Offset Address: 0x00F0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26]	RO	int0_dbg	System bus exception interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[25]	-	reserved	Reserved	0x0
[24]	RO	int0_src0	SRC 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[23:18]	-	reserved	Reserved	0x00
[17]	RO	int0_isp1	ISP 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[16]	RO	int0_isp0	ISP 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15:10]	-	reserved	Reserved	0x00
[9]	RO	int0_ch1	Channel 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	int0_ch0	Channel 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7:2]	-	reserved	Reserved	0x00
[1]	RO	int0_pt1	Port 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	int0_pt0	Port 0 interrupt indicator 0: No interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
			1: An interrupt is generated.	

## VICAP\_INT0\_STATE

VICAP\_INT0\_STATE is an interrupt status register.

Offset Address: 0x00F4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26]	RO	int0_state_dbg	System bus exception interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[25]	-	reserved	Reserved	0x0
[24]	RO	int0_state_src0	SRC 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[23:18]	-	reserved	Reserved	0x00
[17]	RO	int0_state_isp1	ISP 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[16]	RO	int0_state_isp0	ISP 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15:10]	-	reserved	Reserved	0x00
[9]	RO	int0_state_ch1	Channel 1 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	int0_state_ch0	Channel 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7:2]	-	reserved	Reserved	0x00
[1]	RO	int0_state_pt1	Port 1 interrupt indicator	0x0



Bits	Access	Name	Description	Reset
			0: No interrupt is generated. 1: An interrupt is generated.	
[0]	RO	int0_state_pt0	Port 0 interrupt indicator 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## VICAP\_INT0\_MASK

VICAP\_INT0\_MASK is an interrupt mask configuration register.

Offset Address: 0x00F8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26]	RW	mask0_dbg	System bus exception interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[25]	-	reserved	Reserved	0x0
[24]	RW	mask0_src0	SRC 0 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[23:18]	-	reserved	Reserved	0x00
[17]	RW	mask0_isp1	ISP 1 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[16]	RW	mask0_isp0	ISP 0 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[15:10]	-	reserved	Reserved	0x00
[9]	RW	mask0_ch1	Channel 1 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[8]	RW	mask0_ch0	Channel 0 interrupt enable 0: The interrupt is masked.	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
			1: The interrupt is enabled.	
[7:2]	-	reserved	Reserved	0x00
[1]	RW	mask0_pt1	Port 1 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[0]	RW	mask0_pt0	Port 0 interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0

## ONLINE0\_CFG

ONLINE0\_CFG is the online channel 0 control register.

Offset Address: 0x0200 Total Reset Value: 0x0000\_0003

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31]	RW	online0_en	Online output enable 0: disabled 1: enabled	0x0
[30:2]	-	reserved	Reserved	0x00000000
[1]	RW	online0_vsync_mode	Whether to delay the Vsync signal 0: not processed 1: processed	0x1
[0]	RW	online0_hsync_mode	Whether to delay the Hsync signal 0: not processed 1: processed	0x1

## ONLINE0\_SIZE

ONLINE0\_SIZE is the online 0 output active area size register.

Offset Address: 0x02AC Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RO	online0_height	Height of the active image	0x0000



Bits	Access	Name	Description	Reset
[15:0]	RO	online0_width	Width of the active image	0x0000

## VICAP\_AFIFO0\_SIZE

VICAP\_AFIFO0\_SIZE is the VICAP\_AFIFO size configuration register.

Offset Address: 0x030c Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	afifo0_height	Height of the obtained image (in pixels). The configured value is the actual value minus 1.	0x0000
[15:0]	RW	afifo0_width	Width of the obtained image (in pixels). The configured value is the actual value minus 1.	0x0000

## VICAP\_AXI0\_WR\_ERROR

VICAP\_AXI0\_WR\_ERROR is the AXI 0 write error status register.

Offset Address: 0x0fe4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x0
[28]	WC	ch1_tunl_bus_err	Invalid access error of CH1 TUNL Writing <b>1</b> clears this bit.	0x0
[27]	WC	ch0_tunl_bus_err	Invalid access error of CH0 TUNL Writing <b>1</b> clears this bit.	0x0
[26:24]	-	reserved	Reserved	0x0
[23]	WC	isp1_stt_bus_err	Invalid access error of ISP1 STT channel Writing <b>1</b> clears this bit.	0x0
[22]	WC	isp0_stt_bus_err	Invalid access error of ISP0 STT channel Writing <b>1</b> clears this bit.	0x0
[21:4]	-	reserved	Reserved	0x00000
[3]	WC	ch1_c_bus_err	Invalid access error of CH1_C channel	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
			Writing <b>1</b> clears this bit.	
[2]	WC	ch1_y_bus_err	Invalid access error of CH1_Y channel Writing <b>1</b> clears this bit.	0x0
[1]	WC	ch0_c_bus_err	Invalid access error of CH0_C channel Writing <b>1</b> clears this bit.	0x0
[0]	WC	ch0_y_bus_err	Invalid access error of CH0_Y channel Writing <b>1</b> clears this bit.	0x0

## VICAP\_DBG\_INT

VICAP\_DBG\_INT is the system abnormal interrupt status register.

Offset Address: 0x0ff0 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:2]	-	reserved	Reserved	0x00000000
[1]	WC	apb_bus_err	Interrupt status of APB invalid access error  Writing <b>1</b> clears this bit.  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	WC	axi_bus_err	Interrupt status of AXI invalid access error  Writing <b>1</b> clears this bit.  0: No interrupt is generated. 1: An interrupt is generated.	0x0

## VICAP\_DBG\_INT\_MASK

VICAP\_DBG\_INT\_MASK is the system abnormal mask status register.

Offset Address: 0x0ff8 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	apb_bus_err_mask	Interrupt enable for APB invalid access	0x0



Bits	Access	Name	Description	Reset
			error 0: disabled 1: enabled	
[0]	RW	axi_bus_err_mask	Interrupt enable for AXI invalid access error 0: disabled 1: enabled	0x0

## PT\_INTF\_MOD

PT\_INTF\_MOD is the interface mode register.

Offset Address: 0x1000+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31]	RW	pt_enable	0x1000~0x10FF: Register address space of port 0 0x1100~0x11FF: Register address space of port 1 Port enable 0: disabled 1: enabled	0x0
[30:1]	-	reserved	Reserved	0x00000000
[0]	RW	pt_mode	Timing mode 0: external sync mode 1: internal sync	0x0

## PT\_OFFSET0

PT\_OFFSET0 is the component 0 offset register.

Offset Address: 0x1010+PT\_N×0x100 Total Reset Value: 0xFFFF\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	offset0_mask	Component 0 mask	0xFFFF
[15]	RW	offset0_rev	Whether the data line is reversed	0x0



Bits	Access	Name	Description	Reset
			0: No. 1: Yes.	
[14:6]	-	reserved	Reserved	0x000
[5:0]	RW	offset0	Component 0 offset	0x00

## PT\_OFFSET1

PT\_OFFSET1 is the component 1 offset register.

Offset Address: 0x1014+PT\_N×0x100 Total Reset Value: 0xFFFF0\_0010

Bits	Access	Name	Description	Reset
[31:16]	RW	offset1_mask	Component 1 mask	0xFFFF0
[15]	RW	offset1_rev	Whether the data line is reversed 0: No. 1: Yes.	0x0
[14:6]	-	reserved	Reserved	0x000
[5:0]	RW	offset1	Component 1 offset	0x10

## PT\_OFFSET2

PT\_OFFSET2 is the component 2 offset register.

Offset Address: 0x1018+PT\_N×0x100 Total Reset Value: 0xFFFF0\_0020

Bits	Access	Name	Description	Reset
[31:16]	RW	offset2_mask	Component 2 mask	0xFFFF0
[15]	RW	offset2_rev	Whether the data line is reversed 0: No. 1: Yes.	0x0
[14:6]	-	reserved	Reserved	0x000
[5:0]	RW	offset2	Component 2 offset	0x20



## PT\_BT656\_CFG

PT\_BT656\_CFG is the BT.656 configuration register.

Offset Address: 0x1020+PT\_N×0x100 Total Reset Value: 0x0000\_0303

Bits	Access	Name	Description	Reset
[31]	RW	bt656_enable	BT.656 enable 0: disabled 1: enabled	0x0
[30:11]	-	reserved	Reserved	0x00000
[10]	RW	bt656_field_inv	Field inversion control 0: not inverted 1: inverted	0x0
[9]	RW	bt656_vsync_inv	Vsync inversion control 0: not inverted 1: inverted	0x1
[8]	RW	bt656_hsync_inv	Hsync inversion control 0: not inverted 1: inverted	0x1
[7:4]	-	reserved	Reserved	0x0
[3:0]	RW	bt656_mode	Mode select. bt656_mode[0] 0: Hsync is not an active signal. 1: Hsync is the active signal. bt656_mode[1] 0: The Hsync output is active low. 1: The Hsync output is active high. bt656_mode[3:2] 00: Component 0 is parsed. 01: Component 1 is parsed. 10: Component 2 is parsed. 11: reserved	0x3

## PT\_UNIFY\_TIMING\_CFG

PT\_UNIFY\_TIMING\_CFG is the timing configuration register.



Offset Address: 0x1038+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26]	RW	unify_field_inv	Field inversion (level-1 field processing). 0: not inverted 1: inverted	0x0
[25:24]	RW	unify_field_sel	Field source select (level-0 field processing). 00: input field 01: input Vsync 10: detected based on the relationship between Vsync and Hsync 11: fixed at 0	0x0
[23:21]	-	reserved	Reserved	0x0
[20:19]	RW	unify_vsync_mode	Vsync processing mode (level-2 Vsync processing). 00: not processed 01: detect the rising edge 10: detect the rising edge and falling edge 11: reserved	0x0
[18]	RW	unify_vsync_inv	Vsync inversion (level-1 Vsync processing). 0: not inverted 1: inverted	0x0
[17:16]	RW	unify_vsync_sel	Vsync source select (level-0 Vsync processing). 00: input Vsync 01: input field 10: fixed at 0 11: reserved	0x0
[15]	-	reserved	Reserved	0x0
[14:13]	RW	unify_hsync_mode	Hsync processing mode (level-3 Hsync processing). 0: not processed 1: detect the rising edge	0x0



Bits	Access	Name	Description	Reset
[12:11]	RW	unify_hsync_and	Whether Hsync is operated with the result of level-1 Vsync processing (level-2 Hsync processing). 00: not processed 01: ANDed 10: exclusive ORed 11: reserved	0x0
[10]	RW	unify_hsync_inv	Hsync inversion (level-1 Hsync processing). 0: not inverted 1: inverted	0x0
[9:8]	RW	unify_hsync_sel	Hsync source select (level-0 Hsync processing). 00: input Hsync 01: input DE 10: fixed at 0 11: reserved	0x0
[7:3]	-	reserved	Reserved	0x00
[2]	RW	unify_de_inv	de inversion (level-1 de processing). 0: not inverted 1: inverted	0x0
[1:0]	RW	unify_de_sel	de source select (level-0 de processing). 00: input DE 01: result of level-2 Hsync processing 10: fixed at 1 11: fixed at 0	0x0

## PT\_GEN\_TIMING\_CFG

PT\_GEN\_TIMING\_CFG is the timing recovery module configuration register.

Offset Address: 0x103C+PT\_N×0x100 Total Reset Value: 0x4000\_0006

Bits	Access	Name	Description	Reset
[31]	RW	gen_enable	Timing recovery enable (timings are recovered based on timing parameters).	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
			0: disabled 1: enabled	
[30]	RW	gen_mode	Timing recovery mode (timings are recovered based on timing parameters). 0: Timings are generated based on the input valid signal of the port. 1: Timings are automatically calculated and generated internally.	0x1
[29:3]	-	reserved	Reserved	0x00000000
[2]	RW	gen_vsync_mode	Vsync recovery. 0: not recovered 1: recovered	0x1
[1]	RW	gen_hsync_mode	Hsync recovery. 0: not recovered 1: recovered	0x1
[0]	-	reserved	Reserved	0x0

## PT\_UNIFY\_DATA\_CFG

PT\_UNIFY\_DATA\_CFG is the data configuration register.

Offset Address: 0x1040+PT\_N×0x100 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31]	RW	unify_data_enable	Data separation enable 0: disabled 1: enabled	0x0
[30:4]	-	reserved	Reserved	0x00000000
[3]	RW	unify_data_uv_seq	CbCr sequence 0: CbCr 1: CrCb	0x0
[2]	RW	unify_data_yc_seq	YC sequence 0: CY 1: YC	0x0
[1:0]	RW	unify_data_comp_num	Data component select	0x0



Bits	Access	Name	Description	Reset
			00: component 1 01: component 2 10: component 3 11: reserved	

## PT\_GEN\_DATA\_CFG

PT\_GEN\_DATA\_CFG is the data generation module configuration register.

Offset Address: 0x1044+PT\_N×0x100 Total Reset Value: 0x0000\_00E9

Bits	Access	Name	Description	Reset
[31]	RW	gen_data_enable	Data generation enable Data is generated based on the data generation parameter. 0: disabled 1: enabled	0x0
[30:8]	-	reserved	Reserved	0x0000000
[7]	RW	gen_data0_move	Whether data 0 increases progressively 0: No. 1: Yes.	0x1
[6]	RW	gen_data1_move	Whether data 1 increases progressively 0: No. 1: Yes.	0x1
[5]	RW	gen_data2_move	Whether data 2 increases progressively 0: No. 1: Yes.	0x1
[4]	RW	gen_data_vsync_reset	Whether to reset data based on the Vsync signal 0: No. 1: Yes.	0x0
[3]	RW	gen_data_hsync_reset	Whether to reset data based on the Hsync signal 0: No. 1: Yes.	0x1



Bits	Access	Name	Description	Reset
[2]	RW	gen_data_vsync_move	Whether data is incremented progressively based on the Vsync signal 0: No. 1: Yes.	0x0
[1]	RW	gen_data_hsync_move	Whether data is incremented progressively based on the Hsync signal 0: No. 1: Yes.	0x0
[0]	RW	gen_data_de_move	Whether data is incremented progressively based on the de signal 0: No. 1: Yes.	0x1

## PT\_GEN\_DATA\_COEF

PT\_GEN\_DATA\_COEF is the data generation module coefficient register.

Offset Address: 0x1048+PT\_N×0x100 Total Reset Value: 0x0100\_0100

Bits	Access	Name	Description	Reset
[31:24]	RW	inc_frame	Incremental value between data frames The incremental values are accumulated in the upper 8 bits of data.	0x01
[23:16]	RW	step_frame	Interval for data frame increment The configured value is the actual value minus 1. The value <b>0</b> indicates the increment by frame.	0x00
[15:8]	RW	inc_space	Incremental value between data pixels The incremental values are accumulated in the upper 10 bits of data.	0x01
[7:0]	RW	step_space	Interval for data pixel increment The value <b>0</b> indicates the increment by pixel.	0x00



## PT\_GEN\_DATA\_INIT

PT\_GEN\_DATA\_INIT is the data generation initial value configuration register.

Offset Address: 0x104C+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:16]	RW	data2	Initial V/B value	0x00
[15:8]	RW	data1	Initial U/G value	0x00
[7:0]	RW	data0	Initial Y/R value	0x00

## PT\_YUV444\_CFG

PT\_YUV444\_CFG is the YUV444 configuration register.

Offset Address: 0x1050+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31]	RW	yuv44_enable	YUV enable for converting YUV422 signals into YUV444 signals. 0: disabled 1: enabled	0x0
[30:0]	-	reserved	Reserved	0x00000000

## PT\_FSTART\_DLY

PT\_FSTART\_DLY is the port fstart interrupt delay register.

Offset Address: 0x1060+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	pt_fstart_dly	fstart interrupt delay time, in the unit of the port clock	0x00000000

## PT\_FSTART\_H\_DLY

PT\_FSTART\_H\_DLY is the port fstart interrupt horizontal delay register.

Offset Address: 0x1064+PT\_N×0x100 Total Reset Value: 0x0000\_0000



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RW	fstart_h_dly	fstart interrupt line delay time, in the unit of line number of the active area.	0x0000
[15:4]	-	reserved	Reserved	0x000
[3:0]	RW	fstart_h_dly_id	Count frame ID. Bit[0]: Counting is triggered by the timing whose ID is 0. Bit[1]: Counting is triggered by the timing whose ID is 1. Bit[2]: Counting is triggered by the timing whose ID is 2. Bit[3]: Counting is triggered by the timing whose ID is 3.	0x0

## PT\_INTF\_HFB

PT\_INTF\_HFB is the horizontal front blanking width register.

Offset Address: 0x1080+PT\_N×0x100 Total Reset Value: 0x0000\_0058

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	hfb	Width of the horizontal front blanking interval	0x0058

## PT\_INTF\_HACT

PT\_INTF\_HACT is the horizontal active area width register.

Offset Address: 0x1084+PT\_N×0x100 Total Reset Value: 0x0000\_0780

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:0]	RW	hact	Width of the horizontal active area (unit: working clock cycle)	0x00000780

## PT\_INTF\_HBB

PT\_INTF\_HBB is the horizontal back blanking width register.



Offset Address: 0x1088+PT\_N×0x100 Total Reset Value: 0x0000\_00C0

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	hbb	Width of the horizontal back blanking interval	0x00C0

## PT\_INTF\_VFB

PT\_INTF\_VFB is the vertical front blanking width register.

Offset Address: 0x108C+PT\_N×0x100 Total Reset Value: 0x0000\_0004

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	vfb	Width of the vertical front blanking interval	0x0004

## PT\_INTF\_VACT

PT\_INTF\_VACT is the vertical active area width register.

Offset Address: 0x1090+PT\_N×0x100 Total Reset Value: 0x0000\_0438

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	vact	Width of the vertical active area	0x0438

## PT\_INTF\_VBB

PT\_INTF\_VBB is the vertical back blanking width register.

Offset Address: 0x1094+PT\_N×0x100 Total Reset Value: 0x0000\_0029

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	vbb	Width of the vertical back blanking interval	0x0029



## PT\_INTF\_VBFB

PT\_INTF\_VBFB is the vertical bottom front blanking width register.

Offset Address: 0x1098+PT\_N×0x100 Total Reset Value: 0x0000\_0004

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	vbfb	Width of the vertical bottom front blanking interval	0x0004

## PT\_INTF\_VBACT

PT\_INTF\_VBACT is the vertical bottom active area width register.

Offset Address: 0x109C+PT\_N×0x100 Total Reset Value: 0x0000\_0438

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	vbact	Width of the vertical bottom active area	0x0438

## PT\_INTF\_VBBB

PT\_INTF\_VBBB is the vertical bottom back blanking width register.

Offset Address: 0x10A0+PT\_N×0x100 Total Reset Value: 0x0000\_0029

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	vbbb	Width of the vertical bottom back blanking interval	0x0029

## PT\_ID\_CFG

PT\_ID\_CFG is the ID configuration register.

Offset Address: 0x10A4+PT\_N×0x100 Total Reset Value: 0x0000\_0000



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31]	RW	id_enable	ID generation enable 0: disabled 1: enabled	0x0
[30]	RW	id_mode	ID generation mode 0: automatic mode (the ID automatically increases for each frame) 1: non-automatic mode (the ID depends on the configured value)	0x0
[29]	WO	id_reset	In automatic mode, setting this field to 1 restores the ID of the next frame to the initial value. This field is automatically cleared after restoration.	0x0
[28:6]	-	reserved	Reserved	0x0000000
[5:4]	-	reserved	Reserved	0x0
[3:2]	RW	id_max	Maximum ID in automatic mode	0x0
[1:0]	RW	id	Initial ID in automatic mode or generated ID in non-automatic mode	0x0

## PT\_STATUS

PT\_STATUS is the port status register.

Offset Address: 0x10E0+PT\_N×0x100 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:6]	-	reserved	Reserved	0x00000000
[5:4]	RO	pt_status_id	Port output ID	0x0
[3]	RO	pt_status_field	Field output over the port	0x0
[2]	RO	pt_status_vsync	Vsync signal over the port	0x0
[1]	RO	pt_status_hsync	Hsync signal over the port	0x0
[0]	RO	pt_status_de	de signal output over the port	0x0



## PT\_BT656\_STATUS

PT\_BT656\_STATUS is the BT.656 status register.

Offset Address: 0x10E4+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:8]	RO	pt_bt656_seav	Sync code	0x00
[7:0]	-	reserved	Reserved	0x00

## PT\_SIZE

PT\_SIZE is the input image active area size indicator register

Offset Address: 0x10EC+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	pt_height	Height of the active area of the image	0x0000
[15:0]	RO	pt_width	Width of the active area of the image	0x0000

## PT\_INT

PT\_INT is the port interrupt indicator register.

Offset Address: 0x10F0+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3]	WC	pt_fstart_h_dly	Interrupt status of the field/frame start line delay  Writing <b>1</b> clears this bit.  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	WC	pt_height_err	Interrupt status of the image height change  Writing <b>1</b> clears this bit.  0: No interrupt is generated. 1: An interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
[1]	WC	pt_width_err	Interrupt status of the image width change  Writing 1 clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	WC	pt_fstart	Interrupt status of the field/frame start  Writing 1 clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## PT\_INT\_MASK

PT\_INT\_MASK is the port interrupt mask register.

Offset Address: 0x10F8+PT\_N×0x100 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3]	RW	pt_mask_fstart_h_dly	Frame/Field start line delay interrupt enable  0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[2]	RW	pt_mask_height_err	Image height change interrupt enable  0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[1]	RW	pt_mask_width_err	Image width change interrupt enable  0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[0]	RW	pt_mask_fstart	Frame/Field start interrupt enable  0: The interrupt is masked. 1: The interrupt is enabled.	0x0

## CH\_CTRL

CH\_CTRL is the channel control register.



Offset Address: 0x10000+CH\_N×0x1000 Total Reset Value: 0x0002\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31]	RW	ch_enable	0x1_0000~0x1_0FF: register address space of channel 0 0x1_1000~0x1_1FF: register address space of channel 1 Channel enable 0: disabled 1: enabled	0x0
[30]	RW	ch_reg_newer_mode	Enable mode control 0: output when the reg_new register is valid and the module is enabled 1: output when the module is enabled	0x0
[29:18]	-	reserved	Reserved	0x000
[17]	RW	ch_finish_mode	Finish interrupt report time control 0: reported immediately 1: reported at the start of the next frame	0x1
[16:0]	-	reserved	Reserved	0x00000

## CH\_REG\_NEWER

CH\_REG\_NEWER is the capture control register.

Offset Address: 0x10004+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	ch_reg_newer	Channel update register This bit is automatically cleared for each frame.	0x0

## CH\_DLY\_CFG

CH\_DLY\_CFG is the channel input image start interrupt delay configuration register.

Offset Address: 0x10034+CH\_N×0x1000 Total Reset Value: 0x0010\_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	ch_v_dly_cfg	Vertical row delay	0x0010
[15:0]	-	reserved	Reserved	0x0000

## CH\_FRM\_CNT

CH\_FRM\_CNT is the latch frame ID register for a delay interrupt.

Offset Address: 0x10038+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	ch_frm_cnt	Delay interrupt latch frame ID register (0-255)	0x0000
[15:0]	-	reserved	Reserved	0x0000

## CH\_CROP\_CFG

CH\_CROP\_CFG is the channel crop enable register.

Offset Address: 0x10040+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	ch_crop_n0_en	Region 0 enable 0: disabled 1: enabled	0x0

## CH\_CROP0\_START

CH\_CROP0\_START is the channel 0 region crop start register.

Offset Address: 0x10044+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ch_crop_y_start	Start row for obtaining the image	0x0000
[15:0]	RW	ch_crop_x_start	Pixel ID for obtaining the image	0x0000



## CH\_CROP0\_SIZE

CH\_CROP0\_SIZE is the channel 0 region crop size configuration register.

Offset Address: 0x10048+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	ch_crop_height	Height of the obtained image (unit: line). The configured value is the actual value minus 1.	0x0000
[15:0]	RW	ch_crop_width	Width of the obtained image (unit: pixel) The configured value is the actual value minus 1.	0x0000

## CH\_SKIP\_Y\_CFG

CH\_SKIP\_Y\_CFG is the channel Y component skip configuration register.

Offset Address: 0x10050+CH\_N×0x1000 Total Reset Value: 0xFFFF\_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_skipy_cfg	Skip configuration	0xFFFFFFFF

## CH\_SKIP\_C\_CFG

CH\_SKIP\_C\_CFG is the channel C component skip configuration register.

Offset Address: 0x10058+CH\_N×0x1000 Total Reset Value: 0xFFFF\_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RW	ch_skipc_cfg	Skip configuration	0xFFFFFFFF

## CH\_WCH\_Y\_CFG

CH\_WCH\_Y\_CFG is the Y component configuration register.

Offset Address: 0x10080+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31]	RW	wch_y_enable	Channel write enable	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[30]	-	reserved	Reserved	0x0
[29]	RW	wch_y_tunl_en	Tunnel enable	0x0
[28:15]	RW	wch_y_wrap_line	Number of winding buffer lines in the write channel	0x0000
[14]	RW	wch_y_wrap_en	Winding buffer enable register of the write channel 0: disabled 1: enabled	0x0
[13]	RW	wch_y_cmp_mode	Compression mode select 0: frame-based compression 1: line-based compression	0x0
[12]	RW	wch_y_cmp_en	DES channel compression enable 0: disabled 1: enabled	0x0
[11:8]	RW	wch_y_bit_sft	Low alignment of 16-bit data When wch_y_bit_width is set to <b>16</b> , if the input data is 10 bits, the data needs to be shifted by 6 bits.	0x0
[7]	RW	wch_y_bit_sft_mod_e	Shift mode configuration 0: shift of an unsigned number 1: shift of a signed number	0x0
[6]	-	reserved	Reserved	0x0
[5:0]	RW	wch_y_bit_width	Data bit width 0x08: 8 bits 0x0A: 10 bits 0x0C: 12 bits 0x0E: 14 bits 0x10: 16 bits 0x14: 20 bits 0x18: 24 bits 0x1C: 28 bits 0x20: 32 bits	0x00



Bits	Access	Name	Description	Reset
			Other values: reserved	

## CH\_WCH\_Y\_SIZE

CH\_WCH\_Y\_SIZE is the Y component storage size register for the WCH module.

Offset Address: 0x10084+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	wch_y_height	Height of the stored image (unit: row) The configured value is the actual value minus 1.	0x0000
[15:0]	RW	wch_y_width	Width of the stored image (unit: pixel) The configured value is the actual value minus 1.	0x0000

## CH\_WCH\_Y\_OFFSET

CH\_WCH\_Y\_OFFSET is the Y component offset configuration register.

Offset Address: 0x10088+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	wch_y_ver_offset	Vertical offset of the actual image relative to the original image. Only RAW data is supported.	0x0000
[15:0]	-	reserved	Reserved	0x0000

## CH\_WCH\_Y\_FADDR\_L

CH\_WCH\_Y\_FADDR\_L is the lower 32 bits register of the Y component storage base address in the WCH module.

Offset Address: 0x10094+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wch_y_faddr_l	Lower 32 bits of the storage base address of the Y component	0x00000000



## CH\_WCH\_Y\_STRIDE

CH\_WCH\_Y\_STRIDE is the Y component stride register in the WCH module.

Offset Address: 0x100A0+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	wch_y_stride	Stride of the Y component stored in an image, in the unit of byte	0x0000

## CH\_WCH\_C\_CFG

CH\_WCH\_C\_CFG is the C component configuration register in the WCH module.

Offset Address: 0x100B0+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31]	RW	wch_c_enable	CH write channel enable 0: disabled 1: enabled	0x0
[30]	-	reserved	Reserved	0x0
[29]	RW	wch_c_tunl_en	Tunnel enable	0x0
[28:27]	RW	wch_tunl_mode	Interval at which the tunnel information of the OUT0 channel is written 00: written every 2 lines 01: written every 4 lines 10: written every 8 lines 11: written every 16 lines	0x0
[26:13]	RW	wch_tunl_finish_line	Tunnel threshold of the OUT0 channel	0x0000
[12:6]	-	reserved	Reserved	0x00
[5:0]	RW	wch_c_bit_width	Data bit width 0x08: 8 bits 0x0A: 10 bits 0x0C: 12 bits	0x00



Bits	Access	Name	Description	Reset
			0x0E: 14 bits 0x10: 16 bits 0x14: 20 bits 0x18: 24 bits 0x1C: 28 bits 0x20: 32 bits Other values: reserved	

### CH\_WCH\_C\_SIZE

CH\_WCH\_C\_SIZE is the C component storage size register in the WCH module.

Offset Address: 0x100B4+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	wch_c_height	Height of the stored image (unit: row) The configured value is the actual value minus 1.	0x0000
[15:0]	RW	wch_c_width	Width of the stored image (unit: pixel) The configured value is the actual value minus 1.	0x0000

### CH\_WCH\_C\_FADDR\_L

CH\_WCH\_C\_FADDR\_L is the lower 32 bits register of the C component storage base address in the WCH module.

Offset Address: 0x100C4+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	wch_c_faddr_l	Lower 32 bits of the storage base address of the C component	0x00000000

### CH\_WCH\_TUNL\_HADDR\_L

CH\_WCH\_TUNL\_HADDR\_L is the lower 32-bit tunnel base address register.

Offset Address: 0x100CC+CH\_N×0x1000 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	wch_tunl_addr_l	Lower 32 bits of the tunnel base address	0x00000000

## CH\_WCH\_C\_STRIDE

CH\_WCH\_C\_STRIDE is the C component stride register in the WCH module.

Offset Address: 0x100D0+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	wch_c_stride	Stride of the C component of image storage (in bytes)	0x0000

## CH\_Y\_ACT\_SIZE

CH\_Y\_ACT\_SIZE is the output image active area size indicator register of the luminance channel.

Offset Address: 0x100E8+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	ch_y_height_act	Height of the active area of the luminance channel	0x0000
[15:0]	RO	ch_y_width_act	Width of the active area of the luminance channel	0x0000

## CH\_C\_ACT\_SIZE

CH\_C\_ACT\_SIZE is the output image active area size indicator register of the chrominance channel.

Offset Address: 0x100EC+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	ch_c_height_act	Height of the active area of the chrominance channel	0x0000
[15:0]	RO	ch_c_width_act	Width of the active area of the chrominance channel	0x0000



## CH\_INT

CH\_INT is the channel raw interrupt register.

Offset Address: 0x100F0+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	Reserved	0x000
[21]	WC	ch_tunl_line_int	Status of the tunnel threshold interrupt. Writing 1 clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[20]	WC	ch_em_int_buf_ovf	Status of the internal FIFO overflow error interrupt for EM write. Writing 1 clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[19]	WC	ch_stt_int_buf_ovf	Status of the internal FIFO overflow error interrupt for STT write. Writing 1 clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[18]	WC	ch_stt_em_finish	Status of the EM statistics completion interrupt Writing 1 clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[17]	-	reserved	Reserved	0x0
[16]	WC	ch_stt_ae_finish	Status of the AE statistics completion interrupt Writing 1 clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15]	WC	ch_int_fstart_dly	Interrupt status of the frame/field start after delay Writing 1 clears this bit. 0: No interrupt is generated.	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
			1: An interrupt is generated.	
[14:5]	-	reserved	Reserved	0x000
[4]	WC	ch_int_update_cfg	Interrupt status of the register update Writing <b>1</b> clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	WC	ch_int_field_throw	Interrupt status of the frame/field loss Writing <b>1</b> clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	WC	ch_int_buf_ovf	Interrupt status of the internal FIFO overflow error Writing <b>1</b> clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	WC	ch_int_cc_int	Interrupt status of the obtaining completion Writing <b>1</b> clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	WC	ch_int_fstart	Interrupt status of the field/frame start Writing <b>1</b> clears this bit. 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## CH\_INT\_MASK

CH\_INT\_MASK is the channel interrupt mask register.

Offset Address: 0x100F8+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:22]	-	reserved	Reserved	0x000
[21]	RW	mask_tunl_line_int	Tunnel threshold interrupt enable 0: masked	0x0



Bits	Access	Name	Description	Reset
			1: enabled	
[20]	RW	mask_em_buf_ovf	Internal FIFO overflow error interrupt enable for EM write 0: masked 1: enabled	0x0
[19]	RW	mask_stt_buf_ovf	Internal FIFO overflow error interrupt enable for STT write 0: masked 1: enabled	0x0
[18]	RW	mask_stt_em_finis h	Interrupt enable for the EM statistics completion 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[17]	-	reserved	Reserved	0x0
[16]	RW	mask_stt_ae_finish	Interrupt enable for the AE statistics completion 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[15]	RW	mask_fstart_dly	Interrupt enable for field/frame start after delay 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[14:5]	-	reserved	Reserved	0x000
[4]	RW	mask_update_cfg	Register update interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[3]	RW	mask_field_throw	Field/Frame loss interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[2]	RW	mask_buf_ovf	Internal FIFO overflow error interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0
[1]	RW	mask_cc_int	Capture completion interrupt enable	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
			0: The interrupt is masked. 1: The interrupt is enabled.	
[0]	RW	mask_fstart	Field/Frame start interrupt enable 0: The interrupt is masked. 1: The interrupt is enabled.	0x0

### CH\_WCH\_STT\_AE\_HIST\_ADDR\_L

CH\_WCH\_STT\_AE\_HIST\_ADDR\_L is the lower 32 bits register of the ISP AE\_HIST statistics storage base address.

Offset Address: 0x10208+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:0]	RW	ch_wch_stt_ae_hist_addr_l	Higher 32 bits of the ISP AE_HIST statistics storage base address (supported only by channel 0/1)	0x00000000

### CH\_WCH\_STT\_AE\_AVER\_R\_GR\_ADDR\_L

CH\_WCH\_STT\_AE\_AVER\_R\_GR\_ADDR\_L is the lower 32 bits register of the ISP AE\_AVER\_R\_GR statistics storage base address.

Offset Address: 0x10210+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:0]	RW	ch_wch_stt_ae_aver_r_gr_addr_l	Lower 32 bits of the ISP AE_AVER_R_GR statistics storage base address (supported only by channel 0/1)	0x00000000

### CH\_WCH\_STT\_AE\_AVER\_GB\_B\_ADDR\_L

CH\_WCH\_STT\_AE\_AVER\_GB\_B\_ADDR\_L is the lower 32 bits register of the ISP AE\_AVER\_GB\_B statistics storage base address.

Offset Address: 0x10218+CH\_N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:0]	RW	ch_wch_stt_ae_aver_gb_b	Lower 32 bits of the ISP AE_AVER_GB_B	0x00000000



		r_gb_b_addr_l	statistics storage base address (supported only by channel 0/1)	
--	--	---------------	--	--

## 9.2 MIPI Rx

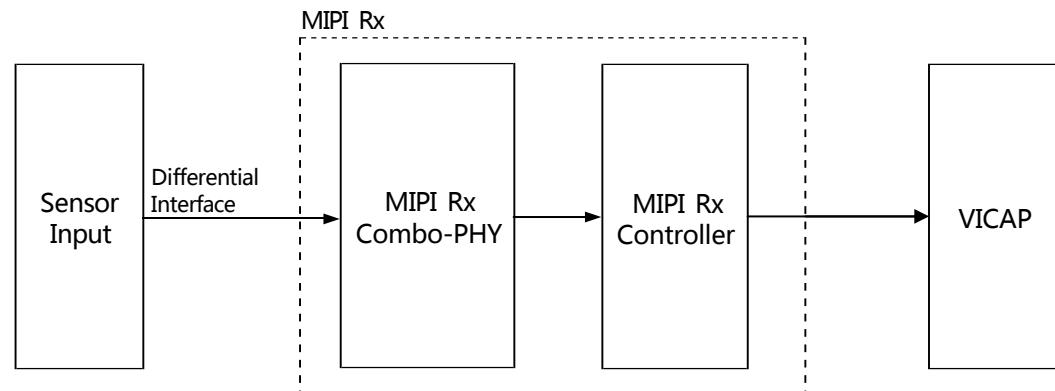
### 9.2.1 Overview

The MIPI RX receives LVDSs that carry raw video data (Bayer RGB data), converts the signals into the digital camera (DC) timing, and transfers it to the downstream VICAP module.

The MIPI RX supports serial video signal inputs such as the MIPI D-PHY, LVDS, and HiSpi. The serial video interfaces provide more bandwidth to enhance transmission stability.

The MIPI RX consists of the Combo-PHY and controller. [Figure 9-17](#) shows the MIPI RX workflow and its position in the system.

**Figure 9-17** MIPI RX workflow and its position in the system



### 9.2.2 Features

#### NOTE

The concepts of lane, link, and channel will appear for multiple times in the following sections.



- A lane is of a differential data pair. At most four data lanes are supported by MIPI\_RX.
- A link refers to a lane group. Each group includes two or four pairs of data. At most two links are supported by MIPI\_RX.
- A channel refers to the data processing channel in the internal MIPI\_RX, which connects to the VICAP channel. One channel independently processes data from one sensor.

The MIPI RX has the following features:

- MIPI DPHY-ver1.1
- Two sensor inputs
- MIPI D-PHY interface with at most four lanes, up to 1.5 Gbit/s per lane
- LVDS/Sub-LVDS/HiSPi interface with at most 4 lanes, up to 1.5 Gbit/s per lane
- Parsing of RAW8/RAW10/RAW12/RAW14/YUV422-8bit/YUV420-8bit nolegacy data
- Endian mode configuration for LVDS or HiSPi mode pixel or synchronization code
- Configuration of the quantity and sequence of lanes

## 9.2.3 Function Description

### 9.2.3.1 Typical Applications

The MIPI RX is a collection unit that supports multiple differential video input interfaces. It is used for converting interface timings. It can receive data through the MIPI, LVDS, sub-LVDS, HiSPi, or DC interface. Depending on the function configuration, the MIPI RX allows data transmission at various rates and transmission of images in various resolutions, and supports multiple image sensors.

The MIPI RX has one D-PHY, with two pairs of differential associated clocks, corresponding to four pairs of data.

Table 9-5 lists the interfaces supported by the MIPI RX.

**Table 9-5** Interfaces supported by the MIPI RX

Interface Type	Common mode voltage	Differential mode voltage	Maximum clock frequency	Maximum data rate per lane
MIPI D-PHY	200 mV	200 mV	750 MHz	1.5 Gbps
sub-LVDS	900 mV	150 mV	750 MHz	1.5 Gbps
LVDS	1.25 V	350 mV	750 MHz	1.5 Gbps
HiSPi(HiVCM)	900 mV	280 mV	750 MHz	1.5 Gbps
HiSPi(SLVS)	200 mV	200 mV	750 MHz	1.5 Gbps



The MIPI RX only converts the interface timings, and does not process image data formats. It supports any resolution and frame rate as long as the bandwidth requirement is met. The MIPI RX bandwidth is limited by the interface data rate and internal processing speed of the Combo PHY. The input interface supports a maximum of 1.5 Gbit/s per lane.

## DNOTE

A maximum rate of 1.5 Gbit/s per lane and simultaneous transmission of at most 4 lanes are supported by the Combo-PHY. For the MIPI RX controller, the channel clock is the same as the VICAP clock.

### 9.2.3.2 Function Implementation

#### Data Format of the MIPI Interface

MIPI specifications are developed and maintained by different working groups and cover diverse application requirements for use in different fields. The MIPI RX supports the D-PHY and CSI-2. D-PHY stipulates the rules of transmission at the physical layer, and CSI-2 stipulates the format and protocol of camera output data packets.

- D-PHY

D-PHY is a high-speed physical layer standard released by the MIPI Alliance. The standard stipulates the physical specifications and transmission protocols at the physical layer for hosts and peripherals. D-PHY uses the LVDS technology with 200 mV source synchronization. The data rate of each lane ranges from 80 Mbit/s to 1500 Mbit/s. D-PHY can work in either low-power (LP) or high-speed (HS) mode.

- CSI-2

CSI-2 is a camera data protocol that stipulates the data packet format used for communication between hosts and peripherals.

CSI-2 supports image application in different pixel formats, and the smallest unit for data transmission is byte. An appropriate number of data lanes can be selected to attain enhanced CSI-2 performance. CSI-2 specifies how the TX end packetizes pixel data into bytes and how multiple data lanes are allocated and managed. Byte data is organized in data packets, which are transmitted between the start of transmission (SoT) and the end of transmission (EoT). The RX end parses the data packets based on the applicable protocol and restores the pixel data.

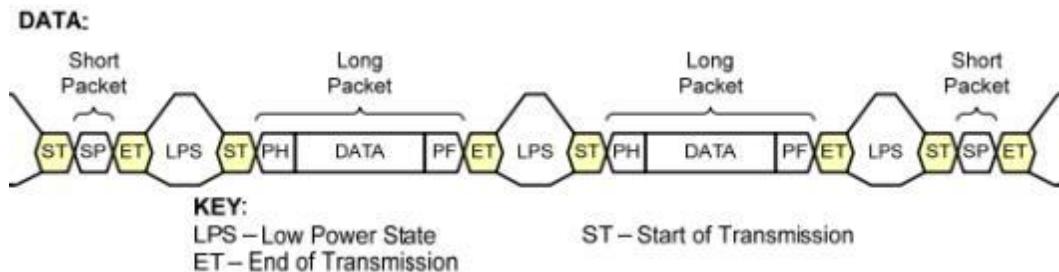
The MIPI RX can parse pixel data in RAW8, RAW10, RAW12, and RAW14 formats.

CSI-2 data packets can be long or short and contain parity codes for parity check and error correction.

Both long and short packets are transmitted between the SoT and EoT. D-PHY works in LP mode in gaps of data transmission. [Figure 9-18](#) shows the mechanism

for transmitting CSI-2 data packets. PH and PF mean packet header and packet footer, respectively.

**Figure 9-18** CSI-2 data packet transmission mechanism

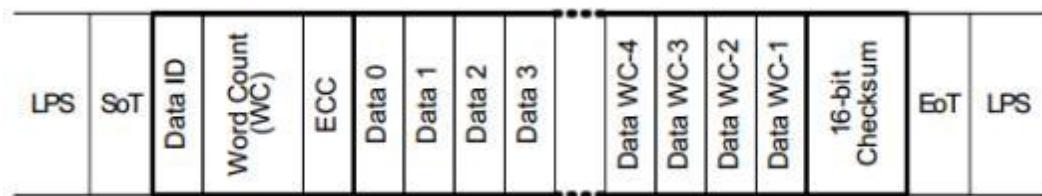


Long packets are used to transmit effective pixel data in five parts: Data ID, Word Count, ECC, Payload, and Checksum.

- Data ID comprises Virtual Channel and Data Type. Virtual Channel controls the channel used for transmission and can specify channel multiplexing so that different channels will transmit different data. Data Type specifies the type of data.
- Word Count indicates the amount of data to be received by the RX end.
- ECC is 8-bit error correction code that can be used to detect and correct errors in Data Type and Word Count.
- Payload is the pixel data to be transmitted.
- Checksum is generated by a linear feedback shift register and is used to check payload data.

[Figure 9-19](#) shows the format of a long packet.

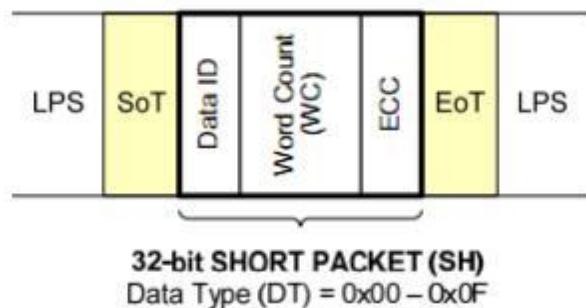
**Figure 9-19** CSI-2 long packet format



Short packets are used to transmit synchronization information in three parts: Data ID, Word Count, and ECC. [Figure 9-20](#) shows its format.



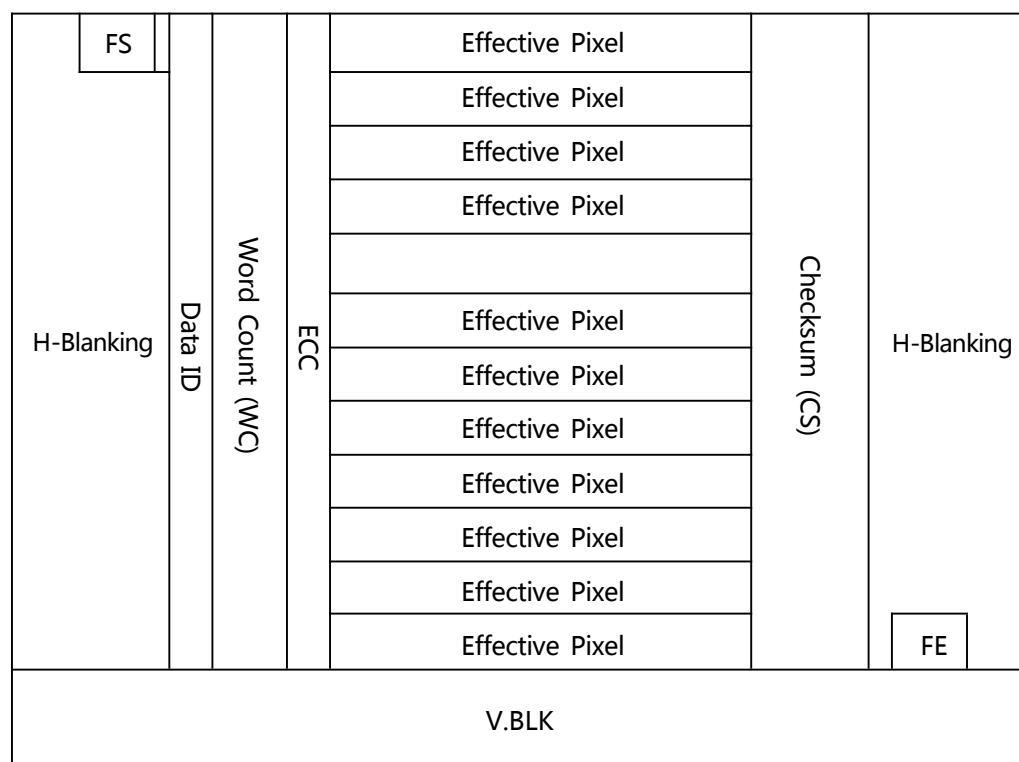
**Figure 9-20** CSI-2 short packet format



## Linear Mode of the MIPI Interface

Figure 9-21 shows the video transfer format in linear mode of the MIPI interface. FS indicates the start of the frame, and FE indicates the end of the frame. The 32-bit data packet in each line consists of the virtual channel and data type information of the current line.

**Figure 9-21** Image data format of the MIPI interface



## Input Data Format of the LVDS Interface

The LVDS is widely used in front-end cameras. It identifies data of the blanking region and active region by synchronization code.



**NOTE**

For the LVDS, there are only the electrical transfer specifications but no standard protocols on the timing and data format. The sub-LVDS is a differential signal technology with ultra-low voltage swing. The common mode/differential mode voltage of the sub-LVDS is lower than that of the LVDS. Compared with the LVDS, the sub-LVDS is more suitable for image sensor applications. The sub-LVDS can be considered as one type of LVDS. In the following sections, sub-LVDS is also called LVDS.

The Combo PHY interface of the MIPI RX converts the differential serial data into parallel data. Then the MIPI RX controller splits and combines the parallel data, extracts the synchronization code, and parses the pixel data.

In LVDS transmission mode, field or line synchronization signals are integrated into data streams. In data streams, SOF and EOF indicate the start and end of a frame, whereas SOL and EOL indicate the start and end of a line. In a data stream, each of SOF, EOF, SOL, and EOL consists of four fields, and the bit width and pixel data of each field are consistent. The first three fields are fixed reference codes, and the fourth field is used to identify the start or end of a frame or line. [Table 9-6](#) illustrates the LVDS synchronization code format.

**Table 9-6** LVDS synchronization code format

Field	Bit Width	Sync code			
		SOF/SAV (Valid line)	EOL/EAV (Valid line)	SOF/SAV (Invalid line)	EOL/EAV (Invalid line)
1st code	8bit	FFh	FFh	FFh	FFh
	10bit	3FFh	3FFh	3FFh	3FFh
	12bit	FFFh	FFFh	FFFh	FFFh
	14bit	3FFFh	3FFFh	3FFFh	3FFFh
	16bit	FFFFh	FFFFh	FFFFh	FFFFh
2nd code	8bit	00h	00h	00h	00h
	10bit	000h	000h	000h	000h
	12bit	000h	000h	000h	000h
	14bit	0000h	0000h	0000h	0000h
	16bit	0000h	0000h	0000h	0000h
3rd code	8bit	00h	00h	00h	00h
	10bit	000h	000h	000h	000h
	12bit	000h	000h	000h	000h
	14bit	0000h	0000h	0000h	0000h



Field	Bit Width	Sync code			
		SOL/SAV (Valid line)	EOL/EAV (Valid line)	SOF/SAV (Invalid line)	EOF/EAV (Invalid line)
	16bit	0000h	0000h	0000h	0000h
4th code	8bit	XXh	XXh	XXh	XXh
	10bit	XXXh	XXXh	XXXh	XXXh
	12bit	XXXh	XXXh	XXXh	XXXh
	14bit	XXXXh	XXXXh	XXXXh	XXXXh
	16bit	XXXXh	XXXXh	XXXXh	XXXXh

#### NOTE

The first three fields of the synchronization code are fixed, and the fourth field identifies the start or end of a field or line. The value of the fourth field is specified by image sensor vendors and therefore it varies according to vendors. [Table 9-7](#) describes an implementation mode.

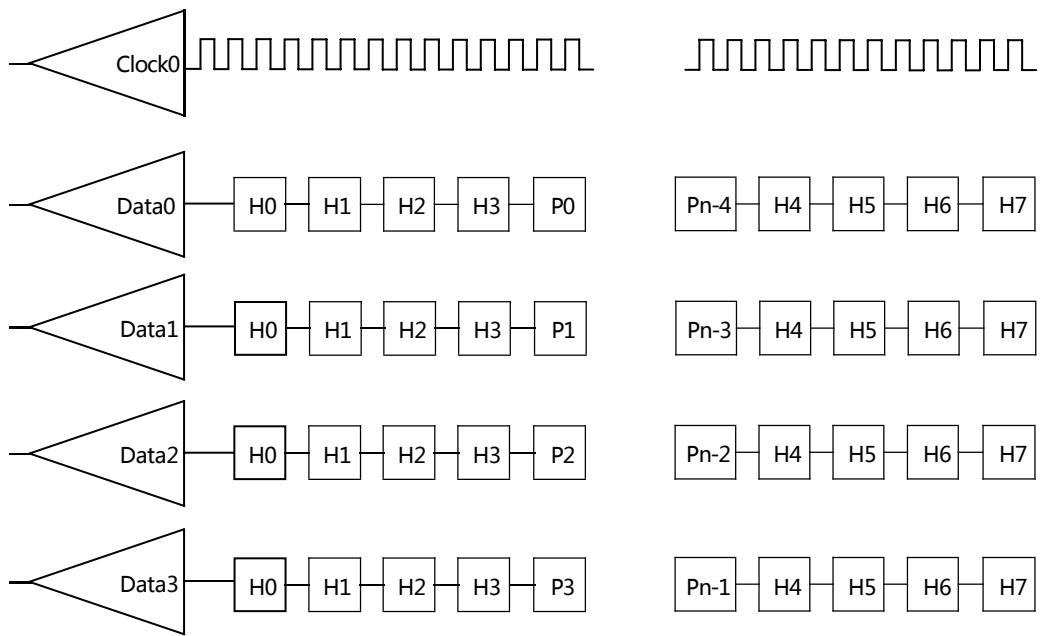
**Table 9-7** Sample of the fourth field of the LVDS synchronization code

Field	Bit Width	Sync code			
		SAV(Valid line)	EAV(Valid line)	SAV(Invalid line)	EAV(Invalid line)
4th code	8bit	80h	9Dh	ABh	B6h
	10bit	200h	274h	2ACh	2D8h
	12bit	800h	9D0h	AB0h	B60h
	14bit	2000h	2740h	2AC0h	2D80h
	16bit	8000h	9D00h	AB00h	B600h

[Figure 9-22](#) shows how LVDS synchronization code and pixel data are transmitted on each of the four lanes. H indicates the synchronization code, and P indicates pixel. The bit widths of H and P are consistent with that of a single output pixel of the image sensor. In each data channel, the synchronization code with four-pixel bit width is transmitted before pixel data. The distribution of pixel data is related to the number of channels. Data is transmitted in series, and the endian mode is configurable in the MIPI RX.

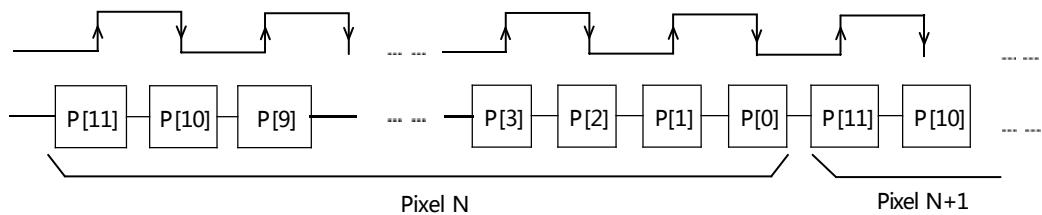


**Figure 9-22** LVDS synchronization code and image transfer mode



The synchronization codes and pixel data are transmitted in serial mode, and the data endian mode is configurable in the MIPI RX.[Figure 9-23](#) shows the timing in which the image sensor outputs a single pixel by taking the RAW12 data and big-endian mode as an example.

**Figure 9-23** LVDS timing of a single pixel



## Linear Mode of the LVDS Interface

Two LVDS synchronization modes are available. In one mode, SAV (Invalid) and EAV (Invalid) identify invalid data of the blanking region whereas SAV (Valid) and EAV (Valid) identify pixel data of the active region.[Figure 9-24](#) shows this synchronization mode.



**Figure 9-24** LVDS synchronization mode 1

H.BLK	SAV (Invalid line)	V.BLK	EAV (Invalid line)	H.BLK
H.BLK		V.BLK		H.BLK
H.BLK		V.BLK		H.BLK
H.BLK	SAV (Valid line)	Effective Pixel	EAV (Valid line)	H.BLK
H.BLK		Effective Pixel		H.BLK
:		:		:
:		:		:
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK	SAV (Invalid line)	V.BLK	EAV (Invalid line)	H.BLK
H.BLK		V.BLK		H.BLK
H.BLK		V.BLK		H.BLK

In the other mode, SOF identifies the start of the first line in the active region, EOF identifies the end of the last line in the active region, and SOL and EOL identify the start and end of the other lines in the active region. [Figure 9-25](#) shows this synchronization mode.



Figure 9-25 LVDS synchronization mode 2

V.BLK				
H.BLK	SOF	Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel	EOL	H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel		H.BLK
H.BLK		Effective Pixel	EOF	H.BLK
V.BLK				

## HiSPi Data Format

HiSPi, formulated by Aptina, includes the HiSPi physical layer protocol and HiSPi specification. The HiSPi physical layer protocol stipulates the electrical specifications and timing parameters, whereas the HiSPi specification defines the data packaging modes.

The HiSPi specification stipulates two physical layer electric standards (HiVCM and SLVS) and four data transmission modes (Packetized-SP, Streaming-SP, Streaming-S, and ActiveStart-SP8).

The MIPI RX supports the HiVCM and SLVS electric standards and supports the Packetized-SP, ActiveStart-SP8, and Streaming-SP mode.

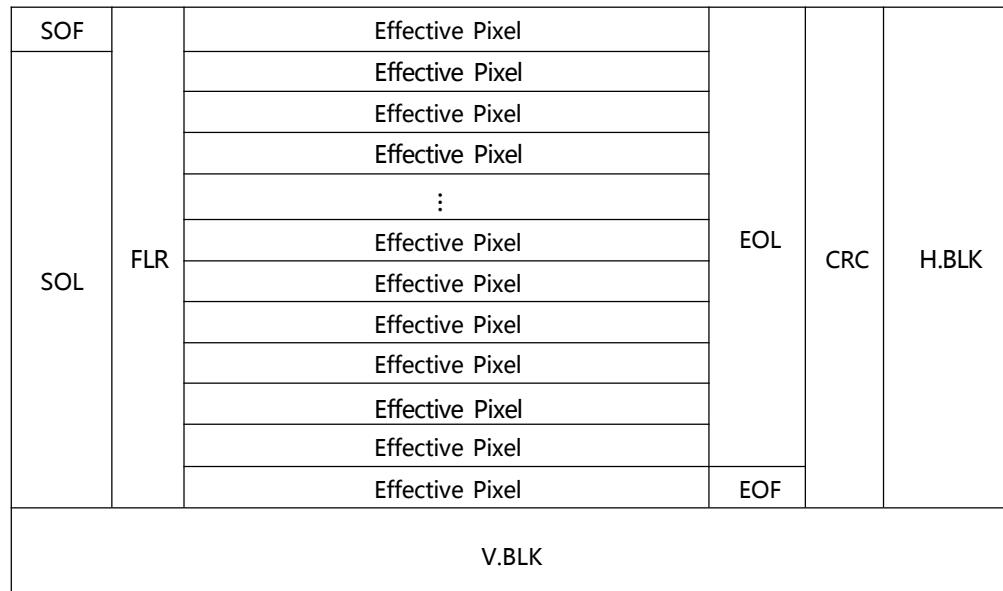
## HiSPi Linear Mode

In Packetized-SP mode, the image sensor identifies the start of the first line in the active region by using the SOF, the end of the last line in the active region by using the EOF, and the start as well end of other lines in the active region by using SOL and EOL respectively. This synchronization method is similar to that in [Figure 9-25](#) except that CRC and FLR can be added to data packets in Packetized-SP mode. [Figure 9-26](#) shows the data format in HiSPi Packetized-SP mode.



The MIPI RX can check whether the matching of SOF-EOF and SOL-EOL is correct. However, it does not process the CRC and FLR data.

**Figure 9-26** Data format in HiSPi Packetized-SP mode



The other two HiSPi transfer modes (Streaming-SP and ActiveStart-SP8) are similar to the Packetized-SP mode except that they differ in the calibration mode of the synchronization code. [Table 9-8](#) describes the differences between the four HiSPi transfer modes. SOF and EOF indicate the start and end of the frame in the active region respectively; SOL and EOL indicate the start and end of the line in the active region respectively; SAV indicates the start of the line in the blanking region.

**Table 9-8** HiSPi transfer modes

Synchronization Code	Packetized-SP	Streaming-SP	ActiveStart-SP8
SOF	Required	Required	Required
SOL	Required	Required	Required
EOF	Required	Unsupported	Unsupported
EOL	Required	Unsupported	Unsupported
SAV	Unsupported	Required	Unsupported

The MIPI RX supports the preceding three modes. All the four fields of a synchronization code can be transmitted in each channel (as shown in [Figure 9-22](#)).



## 9.2.4 Operating Mode of the MIPI RX Controller

The MIPI RX controller supports the MIPI, LVDS, and HiSPi modes. The software configuration in each mode contains two parts: controller and combo-PHY.

### 9.2.4.1 MIPI Mode Configuration Process

In MIPI mode, the required channel quantity, and data type for data transmission as well as PHY operating mode need to be configured. The frame or line synchronization information in MIPI mode is contained in the data packet. The controller parses the data packet and restores the pixel data. The software configuration process in MIPI mode is as follows:

- Step 1** Power on the OS.
- Step 2** Set **mipi0\_work\_mode** and **mipi1\_work\_mode** in the **MIPI\_WORK\_MODE** register to MIPI mode.
- Step 3** Configure the CRG registers PERI\_CRG8528 and PERI\_CRG8536, enable **mipirx\_bus\_cken**, **cil\_cken**, and **mipi\_pix0\_cken**. Configure the soft reset on the MIPI bus and clear the soft reset. Configure the reset on **pix0\_core** of the corresponding channel and clear the reset.
- Step 4** Configure the CRG registers PERI\_CRG8464 and PERI\_CRG8472. Configure the reset on the sensor and clear the reset. Enable sensor clock gating and configure the clock frequency.
- Step 5** Configure the CRG registers PERI\_CRG9305 and PERI\_CRG9313, and select the clock frequency of the **MIPI\_RX** channel.
- Step 6** Configure the number of **MIPI\_RX** lanes (**MIPI\_LANES\_NUM**) and the lane ID (**LANE\_ID\*\_CHN\***).
- Step 7** Configure the received data type and data mode.
- Step 8** Configure the working mode of the PHY (**PHY\_MODE\_LINK\***), PHY channel delay adjustment (**PHY\_SKEW\_LINK\***), PHY channel enable (**PHY\_EN\_LINK\***), and PHY performance adjustment (**PHY\_CFG\_LINK\***).
- Step 9** Configure the system control register. **PHY\_EN**, **LANE\_EN**. Enable **PHY\_CIL\_CTRL** and select **PHYCFG\_MODE** (selecting 0 or 4 in MIPI mode).
- Step 10** Configure **PHYCFG\_EN**.
- Step 11** Configure the sensor sequence.

----End

### 9.2.4.2 LVDS and HiSPi Mode Configuration Process

In LVDS/HiSPi mode, configure the type of raw data, endian mode, synchronization mode and image size registers. In LVDS mode, a synchronization



code is used to identify frame or line synchronization information. Depending on the type of raw data, the synchronization code can be 8, 10, 12 or 14 bits.

The software configuration process in LVDS and HiSPI modes is as follows:

- Step 1** Power on the OS.
- Step 2** Set **mipi0\_work\_mode** and **mipi1\_work\_mode** in the **MIPI\_WORK\_MODE** register to LVDS mode.
- Step 3** Configure the CRG registers PERI\_CRG8528 and PERI\_CRG8536, enable mipirx\_bus\_cken, cil\_cken, and mipi\_pix0\_cken. Configure the soft reset on the MIPI bus and clear the soft reset. Configure the reset on pix0\_core of the corresponding channel and clear the reset.
- Step 4** Configure the CRG registers PERI\_CRG8464 and PERI\_CRG8472. Configure the reset on the sensor and clear the reset. Enable sensor clock gating and configure the clock frequency.
- Step 5** Configure the CRG registers PERI\_CRG9305-PERI and CRG9313, and select the clock frequency of the **MIPI\_Rx** channel.
- Step 6** Configure information including the receive data type and data mode, image width and height, sync header, and lane ID. In LVDS mode, configured width or height value = Actual width or height/Number of lanes - 1
- Step 7** Configure the PHY working mode (**PHY\_MODE\_LINK\***), PHY channel delay adjustment (**PHY\_SKEW\_LINK\***), PHY channel enable (**PHY\_EN\_LINK\***), and PHY performance adjustment (**PHY\_CFG\_LINK\***). Configure the lane sync header (**PHY\_SYNC\_CODE\*\_LINK\***) in LVDS mode.
- Step 8** Configure the system control register. **PHY\_EN**, **LANE\_EN**. Enable **PHY\_CIL\_CTRL** and select **PHYCFG\_MODE** (selecting 1 in LVDS mode).
- Step 9** Configure **PHYCFG\_EN**.
- Step 10** Configure the sensor sequence.

----End

## 9.2.5 MIPI RX Register Summary

[Table 9-9](#)describes the value range and meaning of the variables in the offset addresses for MIPI RX registers.

**Table 9-9** Variables in the MIPI RX register offset addresses

Variable	Value Range	Description
M	0, 1, 2, 3	Lane ID N = 0, the range of M: [0: 3] N = 1, the range of M: [0: 1]



Variable	Value Range	Description
N	0, 1	Port ID

Table 9-10 describes the MIPI RX registers.

**Table 9-10** Summary of MIPI RX registers (base address: 0x173C\_0000)

Offset address	Company name	Description	Page
0x0000	PHY_MODE_LINK	Link PHY operating mode register	<a href="#">9-75</a>
0x0004	PHY_SKEW_LINK	Link PHY channel delay adjustment register	<a href="#">9-77</a>
0x000C	PHY_EN_LINK	Link PHY channel enable register	<a href="#">9-78</a>
0x0018	PHY_DATA_LINK	Link PHY parallel data output register	<a href="#">9-79</a>
0x001C	PHY_PH_MIPI_LINK	Link MIPI packet header register	<a href="#">9-80</a>
0x0020	PHY_DATA_MIPI_LINK	Link data register in MIPI mode	<a href="#">9-80</a>
0x0024	PHY_SYNC_DCT_LINK	Link PHY sync header detection control register in LVDS mode	<a href="#">9-80</a>
0x0030	PHY_SYNC_DOL0_SOF0_LINK	Link PHY sync header register for lane 0 in LVDS mode	<a href="#">9-82</a>
0x0034	PHY_SYNC_DOL0_SOF1_LINK	Link PHY sync header register for lane 1 in LVDS mode	<a href="#">9-82</a>
0x0038	PHY_SYNC_DOL0_SOF2_LINK	Link PHY sync header register for lane 2 in LVDS mode	<a href="#">9-82</a>
0x003C	PHY_SYNC_DOL0_SOF3_LINK	Link PHY sync header register for lane 3 in LVDS mode	<a href="#">9-83</a>
0x0118	PHY_ST2_LINK	Link PHY status 2 register	<a href="#">9-83</a>
0x0150	PHY_FREQ_MEASURE_LINK	Link PHY clock frequency detection module	<a href="#">9-84</a>
0x01F0	MIPI_CIL_INT_RAW_LINK	Link raw interrupt status register	<a href="#">9-84</a>
0x01F4	MIPI_CIL_INT_LINK	Link MIPI CIL interrupt status register	<a href="#">9-85</a>
0x01F8	MIPI_CIL_INT_MSK_LINK	Link MIPI CIL interrupt mask register	<a href="#">9-87</a>
0x0800	HS_MODE_SELECT	HS ID select register	<a href="#">9-89</a>



Offset address	Company name	Description	Page
0x0808	PHY_EN	PHY enable register	<a href="#">9-89</a>
0x080C	LANE_EN	Lane enable register	<a href="#">9-90</a>
0x0810	PHY_CIL_CTRL	PHY CIL control register	<a href="#">9-90</a>
0x0818	PHYCFG_MODE	PHY configuration mode register	<a href="#">9-91</a>
0x081C	PHYCFG_EN	PHY configuration enable register	<a href="#">9-92</a>
0x0824	CHN0_CLR_EN	Channel 0 forcible setting-to-1 enable register	<a href="#">9-92</a>
0x082C	CHN1_CLR_EN	Channel 1 forcible setting-to-1 enable register	<a href="#">9-93</a>
0x0FF0	MIPI_INT_RAW	MIPI system raw interrupt status register	<a href="#">9-93</a>
0x0FF4	MIPI_INT_ST	MIPI system interrupt status register	<a href="#">9-94</a>
0x0FF8	MIPI_INT_MSK	MIPI system interrupt mask register	<a href="#">9-94</a>
0x1020+N×0x100	MIPI(N)_CRC_INTR_RAW	MIPI (N) CSI raw interrupt status register	<a href="#">9-95</a>
0x1024+N×0x100	MIPI(N)_CRC_INTR_ST	MIPI (N) CSI masked interrupt status register	<a href="#">9-96</a>
0x1028+N×0x100	MIPI(N)_CRC_INTR_MSK	MIPI (N) CSI interrupt mask register	<a href="#">9-98</a>
0x1100+N×0x100	MIPI(N)_USERDEF_DT	MIPI (N) pixel bit width configuration register for the user-defined data type	<a href="#">9-100</a>
0x1104+N×0x100	MIPI(N)_USER_DEF	MIPI (N) configuration enable register for the user-defined data type	<a href="#">9-100</a>
0x1108+N×0x100	MIPI(N)_CTRL_MODE_HS	MIPI (N) operating mode enable register	<a href="#">9-101</a>
0x1200+N×0x100	MIPI(N)_DOL_ID_CODE0	MIPI (N) DOL mode frame identification 0 register	<a href="#">9-101</a>
0x1204+N×0x100	MIPI(N)_DOL_ID_CODE1	MIPI (N) DOL mode frame identification 1 register	<a href="#">9-102</a>
0x1208+N×0x100	MIPI(N)_DOL_ID_CODE2	MIPI (N) DOL mode frame identification 2 register	<a href="#">9-102</a>



Offset address	Company name	Description	Page
0x1210+N×0x100	MIPI(N)_CROP_START_CHN0	Size register for the start point of MIPI (N) channel 0 cropping	<a href="#">9-102</a>
0x1214+N×0x100	MIPI(N)_CROP_START_CHN1	Size register for the start point of MIPI (N) channel 1 cropping	<a href="#">9-103</a>
0x1224+N×0x100	MIPI(N)_IMGSIZE	MIPI (N) image size register	<a href="#">9-103</a>
0x1230+N×0x100	MIPI(N)_CTRL_MODE_PIXEL	MIPI (N) output operating mode enable register	<a href="#">9-103</a>
0x1240+N×0x100	MIPI(N)_DUMMY_PIX_REG	MIPI (N) dummy line pixel value register	<a href="#">9-105</a>
0x1250+N×0x100	MIPI(N)_IMGSIZE0_STATIS	MIPI (N) VC 0 transferred image size register	<a href="#">9-105</a>
0x1254+N×0x100	MIPI(N)_IMGSIZE1_STATIS	MIPI (N) VC 1 transferred image size register	<a href="#">9-105</a>
0x12F0+N×0x100	MIPI(N)_CTRL_INT_RAW	MIPI (N) read data error raw interrupt status register	<a href="#">9-106</a>
0x12F4+N×0x100	MIPI(N)_CTRL_INT	MIPI (N) read data error interrupt status register	<a href="#">9-106</a>
0x12F8+N×0x100	MIPI(N)_CTRL_INT_MSK	MIPI (N) read data error interrupt mask register	<a href="#">9-107</a>
0x1300+N×0x100	LVDS(N)_WDR	LVDS (N) WDR control register	<a href="#">9-108</a>
0x1304+N×0x100	LVDS(N)_DOLSCD_HBLK	LVDS (N) SCD control register	<a href="#">9-110</a>
0x1308+N×0x100	LVDS(N)_CTRL	LVDS (N) control register	<a href="#">9-110</a>
0x130C+N×0x100	LVDS(N)_IMGSIZE	LVDS (N) image size register	<a href="#">9-111</a>
0x1310+N×0x100	LVDS(N)_CROP_START0	Cropping register for the first LVDS (N) frame	<a href="#">9-112</a>
0x1314+N×0x100	LVDS(N)_CROP_START1	Cropping register for the second LVDS (N) frame.	<a href="#">9-112</a>
0x1320+M×0x20+N×0x1000	LVDS(N)_LANE(M)_SOF_01	SOF sync code configuration register for lane (M) in LVDS or HiSPI mode	<a href="#">9-112</a>



Offset address	Company name	Description	Page
0x1328+M×0x20+N×0x1000	LVDS(N)_LANE(M)_EOF_01	EOF sync code configuration register for lane (M) in LVDS or HiSPI mode	<a href="#">9-113</a>
0x1330+M×0x20+N×0x1000	LVDS(N)_LANE(M)_SOL_01	SOL sync code configuration register for lane (M) in LVDS or HiSPI mode	<a href="#">9-113</a>
0x1338+M×0x20+N×0x1000	LVDS(N)_LANE(M)_EOL_01	EOL sync code configuration register for lane (M) in LVDS or HiSPI mode	<a href="#">9-114</a>
0x1520+M×0x20+N×0x1000	LVDS(N)_LANE(M)_NXT_SOF_01	Lane (M) SOF sync code configuration register for frame (N + 1) in LVDS or HiSPI mode	<a href="#">9-114</a>
0x1528+M×0x20+N×0x1000	LVDS(N)_LANE(M)_NXT_EOF_01	Lane (M) EOF sync code configuration register for frame (N + 1) in LVDS or HiSPI mode	<a href="#">9-115</a>
0x1530+M×0x20+N×0x1000	LVDS(N)_LANE(M)_NXT_SOL_01	Lane (M) SOL sync code configuration register for frame (N + 1) in LVDS or HiSPI mode	<a href="#">9-115</a>
0x1538+M×0x20+N×0x1000	LVDS(N)_LANE(M)_NXT_EOL_01	Lane (M) EOL sync code configuration register for frame (N + 1) in LVDS or HiSPI mode	<a href="#">9-116</a>
0x1720+N×0x100	LVDS(N)_LI_WORD0	LVDS DOL mode frame 0 LI register	<a href="#">9-116</a>
0x1724+N×0x100	LVDS(N)_LI_WORD1	LVDS DOL mode frame 1 LI register	<a href="#">9-117</a>
0x1780+N×0x100	LVDS(N)_IMGSIZE0_STATIS	LVDS LEF image size statistics register	<a href="#">9-117</a>
0x1784+N×0x100	LVDS(N)_IMGSIZE1_STATIS	LVDS SEF 1 image size statistics register	<a href="#">9-117</a>
0x17A0+N×0x100	LVDS(N)_OUTPUT_PIX_NUM	LVDS output pixel bit width selection register	<a href="#">9-117</a>
0x17F0+N×0x100	LVDS(N)_CTRL_INT_RAW	LVDS read data raw interrupt status register	<a href="#">9-118</a>
0x17F4+N×0x100	LVDS(N)_CTRL_INT	LVDS read data interrupt status register	<a href="#">9-119</a>
0x17F8+N×0x100	LVDS(N)_CTRL_INT_MSK	LVDS read data interrupt mask register	<a href="#">9-120</a>



Offset address	Company name	Description	Page
0x1800+N×0x1000	LANE_ID0_CHN(N)	Lane priority configuration register 0	<a href="#">9-122</a>
0x18F0+N×0x1000	ALIGN(N)_INT_RAW	MIPI_ALIGN raw interrupt status register	<a href="#">9-122</a>
0x18F4+N×0x1000	ALIGN(N)_INT	MIPI_ALIGN interrupt status register	<a href="#">9-123</a>
0x18F8+N×0x1000	ALIGN(N)_INT_MSK	MIPI_ALIGN interrupt mask register	<a href="#">9-124</a>
0x1FF0+N×0x1000	CHN(N)_INT_RAW	Channel (N) raw interrupt status register	<a href="#">9-124</a>
0x1FF4+N×0x1000	CHN(N)_INT	Channel (N) interrupt status register	<a href="#">9-125</a>
0x1FF8+N×0x1000	CHN(N)_INT_MASK	Channel (N) interrupt mask register	<a href="#">9-126</a>

## 9.2.6 MIPI RX Register Description

### PHY\_MODE\_LINK

PHY\_MODE\_LINK is the link PHY operating mode register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25]	RW	phy0_rg_mipi_mod_e1	MIPI/LVDS input mode select. When the PHY works in LVDS mode and the common voltage is greater than or equal to 1200 mV, set this bit to 0. 0: The common voltage is greater than 1200 mV. 1: The common voltage is 1200 mV or below.	0x0
[24]	RW	phy0_rg_mipi_mod_e0	MIPI/LVDS input mode select. When the PHY works in LVDS mode and the common voltage is greater than or equal to 900 mV, set this bit to 0. 0: The common voltage is greater than	0x0



Bits	Access	Name	Description	Reset
			1200 mV. 1: The common voltage is 1200 mV or below.	
[23:21]	-	reserved	Reserved	0x0
[20]	RW	phy0_rg_en_2l2l	Differential clock source select 0: A pair of differential clocks is input externally. 1: Two pairs of differential clocks are input externally.	0x0
[19:18]	-	reserved	Reserved	0x0
[17]	RW	phy0_rg_faclk1_en	Associated clock 1 phase for the PHY output data 0: Data is output on the rising edge of the clock. 1: Data is output on the falling edge of the clock.	0x0
[16]	RW	phy0_rg_faclk0_en	Associated clock 0 phase for the PHY output data 0: Data is output on the rising edge of the clock. 1: Data is output on the falling edge of the clock.	0x0
[15:14]	-	reserved	Reserved	0x0
[13]	RW	phy0_rg_en_lp1	LP1 mode enable. This mode is enabled in MIPI mode and disabled in other modes. 0: The LP1 mode of the PHY is disabled. 1: The LP1 mode of the PHY is enabled.	0x0
[12]	RW	phy0_rg_en_lp0	LP0 mode enable. This mode is enabled in MIPI mode and disabled in other modes. 0: The LP0 mode of the PHY is disabled. 1: The LP0 mode of the PHY is enabled.	0x0
[11:9]	-	reserved	Reserved	0x0
[8]	RW	phy0_rg_en_cmos	CMOS mode enable 0: The CMOS mode of the PHY is	0x0



Bits	Access	Name	Description	Reset
			disabled. 1: The CMOS mode of the PHY is enabled.	
[7:6]	-	reserved	Reserved	0x0
[5]	RW	phy0_rg_en_clk1	Clock 1 lane enable 0: disabled 1: enabled	0x0
[4]	RW	phy0_rg_en_clk0	Clock 0 lane enable 0: disabled 1: enabled	0x0
[3:0]	RW	phy0_rg_en_d	Data lane enable 0: The data lane is disabled. 1: The data lane is enabled.	0x0

## PHY\_SKew\_LINK

PHY\_SKew\_LINK is the link PHY channel delay adjustment register.

Offset Address: 0x0004 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	Reserved	0x000
[22:20]	RW	phy0_d3_skew	Timing delay adjustment of data lane 3. Each time the value of this register is increased by 1, the phase is delayed by about 62.5 ps.	0x0
[19]	-	reserved	Reserved	0x0
[18:16]	RW	phy0_d2_skew	Timing delay adjustment of data lane 2. Each time the value of this register is increased by 1, the phase is delayed by about 62.5 ps.	0x0
[15]	-	reserved	Reserved	0x0
[14:12]	RW	phy0_d1_skew	Timing delay adjustment of data lane 1. Each time the value of this register is increased by 1, the phase is delayed by about 62.5 ps.	0x0



Bits	Access	Name	Description	Reset
[11]	-	reserved	Reserved	0x0
[10:8]	RW	phy0_d0_skew	Timing delay adjustment of data lane 0. Each time the value of this register is increased by 1, the phase is delayed by about 62.5 ps.	0x0
[7]	-	reserved	Reserved	0x0
[6:4]	RW	phy0_clk1_skew	Timing delay adjustment of clock lane 1. Each time the value of this register is increased by 1, the phase is delayed by about 62.5 ps.	0x0
[3]	-	reserved	Reserved	0x0
[2:0]	RW	phy0_clk0_skew	Timing delay adjustment of clock lane 0. Each time the value of this register is increased by 1, the phase is delayed by about 62.5 ps.	0x0

## PHY\_EN\_LINK

PHY\_EN\_LINK is the link PHY channel enable register.

Offset Address: 0x000C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13]	RW	phy_clk1_term_en	Termination impedance matching enable for clock 1 lane 0: disabled 1: enabled	0x0
[12]	RW	phy_clk0_term_en	Termination impedance matching enable for clock 0 lane 0: disabled 1: enabled	0x0
[11]	RW	phy_d3_term_en	Termination impedance matching enable for data lane 3 0: disabled 1: enabled	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[10]	RW	phy_d2_term_en	Termination impedance matching enable for data lane 2 0: disabled 1: enabled	0x0
[9]	RW	phy_d1_term_en	Termination impedance matching enable for data lane 1 0: disabled 1: enabled	0x0
[8]	RW	phy_d0_term_en	Termination impedance matching enable for data lane 0 0: disabled 1: enabled	0x0
[7:4]	-	reserved	Reserved	0x0
[3]	RW	phy_da_d3_valid	High-speed mode enable for data lane 3 0: disabled 1: enabled	0x0
[2]	RW	phy_da_d2_valid	High-speed mode enable for data lane 2 0: disabled 1: enabled	0x0
[1]	RW	phy_da_d1_valid	High-speed mode enable for data lane 1 0: disabled 1: enabled	0x0
[0]	RW	phy_da_d0_valid	High-speed mode enable for data lane 0 0: disabled 1: enabled	0x0

## PHY\_DATA\_LINK

PHY\_DATA\_LINK is the link PHY parallel data output register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:24]	RO	phy_data3_mipi	Data received by data lane 3	0x00
[23:16]	RO	phy_data2_mipi	Data received by data lane 2	0x00



[15:8]	RO	phy_data1_mipi	Data received by data lane 1	0x00
[7:0]	RO	phy_data0_mipi	Data received by data lane 0	0x00

## PHY\_PH\_MIPI\_LINK

PHY\_PH\_MIPI\_LINK is the link MIPI packet header register.

Offset Address: 0x001C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	phy_ph3_mipi	Packet header received by data lane 3	0x00
[23:16]	RO	phy_ph2_mipi	Packet header received by data lane 2	0x00
[15:8]	RO	phy_ph1_mipi	Packet header received by data lane 1	0x00
[7:0]	RO	phy_ph0_mipi	Packet header received by data lane 0	0x00

## PHY\_DATA\_MIPI\_LINK

PHY\_DATA\_MIPI\_LINK is the link data register in MIPI mode.

Offset Address: 0x0020 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	RO	phy_data3_mipi_hs	Data in MIPI format in data lane 3	0x00
[23:16]	RO	phy_data2_mipi_hs	Data in MIPI format in data lane 2	0x00
[15:8]	RO	phy_data1_mipi_hs	Data in MIPI format in data lane 1	0x00
[7:0]	RO	phy_data0_mipi_hs	Data in MIPI format in data lane 0	0x00

## PHY\_SYNC\_DCT\_LINK

PHY\_SYNC\_DCT\_LINK is the link PHY sync header detection control register in LVDS mode

Offset Address: 0x0024 Total Reset Value: 0x0010\_1101

Bits	Access	Name	Description	Reset
[31:21]	-	reserved	Reserved	0x000



Bits	Access	Name	Description	Reset
[20]	RW	cil_code_big_endian1	Serial bit transmission sequence 1 for the sync code (sync_code) of the raw data to be transmitted in LVDS or HiSPi mode  0: LSBs are transmitted first. The serial data sequence of the sync_code received is bit 0, bit 1... bit 11. 1: MSBs are transmitted first. The serial data sequence of the sync_code received is bit 11, bit 10... bit 0.	0x1
[19:15]	-	reserved	Reserved	0x00
[14:12]	RW	cil_raw_type1	Type 1 of raw data to be transmitted in LVDS or HiSPi mode  001: 8-bit raw data 010: 10-bit raw data 011: 12-bit raw data 100: 14-bit raw data Other values: reserved	0x1
[11:9]	-	reserved	Reserved	0x0
[8]	RW	cil_code_big_endian0	Serial bit transmission sequence 0 for the sync_code of raw data to be transmitted in LVDS or HiSPi mode  0: LSBs are transmitted first. The serial data sequence of the sync_code received is bit 0, bit 1... bit 11. 1: MSBs are transmitted first. The serial data sequence of the sync_code received is bit 11, bit 10... bit 0.	0x1
[7:3]	-	reserved	Reserved	0x00
[2:0]	RW	cil_raw_type0	Type 0 of raw data to be transmitted in LVDS or HiSPi mode  001: 8-bit raw data 010: 10-bit raw data 011: 12-bit raw data 100: 14-bit raw data Other values: reserved	0x1



## PHY\_SYNC\_DOL0\_SOFO\_LINK

PHY\_SYNC\_DOL0\_SOFO\_LINK is the link PHY sync header register for lane 0 in LVDS mode.

Offset Address: 0x0030 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	cil_u0lvds_sof1_wo_rd4_0	SOF sync code for lane 0 ( $N+1$ frames)	0x0000
[15:0]	RW	cil_u0lvds_sof0_wo_rd4_0	SOF sync code for lane 0 ( $N$ frames)	0x0000

## PHY\_SYNC\_DOL0\_SOFR\_LINK

PHY\_SYNC\_DOL0\_SOFR\_LINK is the link PHY sync header register for lane 1 in LVDS mode.

Offset Address: 0x0034 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	cil_u1lvds_sof1_wo_rd4_0	SOF sync code for lane 1 ( $N+1$ frames)	0x0000
[15:0]	RW	cil_u1lvds_sof0_wo_rd4_0	SOF sync code for lane 1 ( $N$ frames)	0x0000

## PHY\_SYNC\_DOL0\_SOFR2\_LINK

PHY\_SYNC\_DOL0\_SOFR2\_LINK is the link PHY sync header register for lane 2 in LVDS mode.

Offset Address: 0x0038 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	cil_u2lvds_sof1_wo_rd4_0	SOF sync code for lane 2 ( $N+1$ frames)	0x0000
[15:0]	RW	cil_u2lvds_sof0_wo_rd4_0	SOF sync code for lane 2 ( $N$ frames)	0x0000



## PHY\_SYNC\_DOL0\_SOF3\_LINK

PHY\_SYNC\_DOL0\_SOF3\_LINK is the link PHY sync header register for lane 3 in LVDS mode.

Offset Address: 0x003C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	cil_u3lvds_sof1_wo_rd4_0	SOF sync code for lane 3 ( $N+1$ frames)	0x0000
[15:0]	RW	cil_u3lvds_sof0_wo_rd4_0	SOF sync code for lane 3 ( $N$ frames)	0x0000

## PHY\_ST2\_LINK

PHY\_ST2\_LINK is the link PHY status 2 register.

Offset Address: 0x0118 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:14]	RO	reserved	Reserved	0x00000
[13]	RO	physt_clk1_term_en	Termination matched impedance enable for clock lane 1 0: disabled 1: enabled	0x0
[12]	RO	physt_clk0_term_en	Termination matched impedance enable for clock lane 0 0: disabled 1: enabled	0x0
[11]	RO	physt_d3_term_en	Termination matched impedance enable for data lane 3 0: disabled 1: enabled	0x0
[10]	RO	physt_d2_term_en	Termination matched impedance enable for data lane 2 0: disabled 1: enabled	0x0
[9]	RO	physt_d1_term_en	Termination matched impedance enable for data lane 1	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[8]	RO	physt_d0_term_en	Termination matched impedance enable for data lane 0 0: disabled 1: enabled	0x0
[7:0]	-	reserved	Reserved	0x0

## PHY\_FREQ\_MEASURE\_LINK

PHY\_FREQ\_MEASURE\_LINK is the link PHY clock frequency detection module.

Offset Address: 0x0150 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	cil_freq_meas_res	Detection value of the frequency detection module	0x00000000

## MIPI\_CIL\_INT\_RAW\_LINK

MIPI\_CIL\_INT\_RAW\_LINK is the link raw interrupt status register.

Offset Address: 0x01F0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12]	WC	err0_timeout_ck0_raw	Raw FSM timeout interrupt status for clock0 lane 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[11]	WC	err0_timeout_d3_raw	Raw FSM timeout interrupt status for data lane 3 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[10]	WC	err0_timeout_d2_raw	Raw FSM timeout interrupt status for data lane 2 0: There is no raw interrupt.	0x0



Bits	Access	Name	Description	Reset
			1: There is a raw interrupt.	
[9]	WC	err0_timeout_d1_rw	Raw FSM timeout interrupt status for data lane 1 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[8]	WC	err0_timeout_d0_rw	Raw FSM timeout interrupt status for data lane 0 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[7:5]	-	reserved	Reserved	0x0
[4]	WC	err0_escape_ck0_rw	Raw escape sequence interrupt status for clock0 lane 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[3]	WC	err0_escape_d3_rw	Raw escape sequence interrupt status for data lane 3 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[2]	WC	err0_escape_d2_rw	Raw escape sequence interrupt status for data lane 2 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[1]	WC	err0_escape_d1_rw	Raw escape sequence interrupt status for data lane 1 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[0]	WC	err0_escape_d0_rw	Raw escape sequence interrupt status for data lane 0 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0

## MIPI\_CIL\_INT\_LINK

MIPI\_CIL\_INT\_LINK is the link MIPI CIL interrupt status register.

Offset Address: 0x01F4 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13]	RO	err0_timeout_ck1_st	FSM timeout interrupt status for clock1 lane 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[12]	RO	err0_timeout_ck0_st	FSM timeout interrupt status for clock0 lane 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[11]	RO	err0_timeout_d3_st	FSM timeout interrupt status for data lane 3 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[10]	RO	err0_timeout_d2_st	FSM timeout interrupt status for data lane 2 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[9]	RO	err0_timeout_d1_st	FSM timeout interrupt status for data lane 1 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	err0_timeout_d0_st	FSM timeout interrupt status for data lane 0 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7:6]	-	reserved	Reserved	0x0
[5]	RO	err0_escape_ck1_st	Escape sequence interrupt status for clock1 lane 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	err0_escape_ck0_st	Escape sequence interrupt status for clock0 lane 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	RO	err0_escape_d3_st	Escape sequence interrupt status for	0x0



Bits	Access	Name	Description	Reset
			clock lane 3 0: No interrupt is generated. 1: An interrupt is generated.	
[2]	RO	err0_escape_d2_st	Escape sequence interrupt status for clock lane 2 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	err0_escape_d1_st	Escape sequence interrupt status for clock lane 1 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	err0_escape_d0_st	Escape sequence interrupt status for clock lane 0 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## **MIPI\_CIL\_INT\_MSK\_LINK**

MIPI\_CIL\_INT\_MSK\_LINK is the link MIPI CIL interrupt mask register.

Offset Address: 0x01F8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x00000
[13]	RW	err3_timeout_ck1_ms_k	FSM timeout interrupt enable for clock1 lane 0: disabled 1: enabled	0x0
[12]	RW	err3_timeout_ck0_ms_k	FSM timeout interrupt enable for clock0 lane 0: disabled 1: enabled	0x0
[11]	RW	err3_timeout_d3_ms_k	FSM timeout interrupt enable for data lane 3 0: disabled 1: enabled	0x0



Bits	Access	Name	Description	Reset
[10]	RW	err3_timeout_d2_ms_k	FSM timeout interrupt enable for data lane 2 0: disabled 1: enabled	0x0
[9]	RW	err3_timeout_d1_ms_k	FSM timeout interrupt enable for data lane 1 0: disabled 1: enabled	0x0
[8]	RW	err3_timeout_d0_ms_k	FSM timeout interrupt enable for data lane 0 0: disabled 1: enabled	0x0
[7:6]	-	reserved	Reserved	0x0
[5]	RW	err3_escape_ck1_ms_k	Escape sequence interrupt enable for clock1 lane 0: disabled 1: enabled	0x0
[4]	RW	err3_escape_ck0_ms_k	Escape sequence interrupt enable for clock0 lane 0: disabled 1: enabled	0x0
[3]	RW	err3_escape_d3_msk	Escape sequence interrupt enable for clock lane 3 0: disabled 1: enabled	0x0
[2]	RW	err3_escape_d2_msk	Escape sequence interrupt enable for clock lane 2 0: disabled 1: enabled	0x0
[1]	RW	err3_escape_d1_msk	Escape sequence interrupt enable for clock lane 1 0: disabled 1: enabled	0x0
[0]	RW	err3_escape_d0_msk	Escape sequence interrupt enable for clock lane 0	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	

## HS\_MODE\_SELECT

HS\_MODE\_SELECT is the HS ID select register.

Offset Address: 0x0800 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3:0]	RW	hs_mode	Scenario mode select 0x0: 8Lane 0x7: 4Lane+4Lane 0xA: 4Lane+2Lane+2Lane 0xB: 2Lane+2Lane+2Lane+2Lane Other values: reserved	0x0

## PHY\_EN

PHY\_EN is the PHY enable register.

Offset Address: 0x0808 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	Reserved	0x00000000
[8]	RO	reserved	Reserved	0x0
[7]	RW	d_p_swap_data3	Differential PN swap enable for lane 3 0: disabled 1: enabled	0x0
[6]	RW	d_p_swap_data2	Differential PN swap enable for lane 2 0: disabled 1: enabled	0x0
[5]	RW	d_p_swap_data1	Differential PN swap enable for lane 1 0: disabled	0x0



Bits	Access	Name	Description	Reset
			1: enabled	
[4]	RW	d_p_swap_data0	Differential PN swap enable for lane 0 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	phy0_en	PHY 0 enable 0: disabled 1: enabled	0x0

## LANE\_EN

LANE\_EN is the lane enable register.

Offset Address: 0x080C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3]	RW	lane3_en	Lane 3 enable 0: disabled 1: enabled	0x0
[2]	RW	lane2_en	Lane 2 enable 0: disabled 1: enabled	0x0
[1]	RW	lane1_en	Lane 1 enable 0: disabled 1: enabled	0x0
[0]	RW	lane0_en	Lane 0 enable 0: disabled 1: enabled	0x0

## PHY\_CIL\_CTRL

PHY\_CIL\_CTRL is the PHY CIL control register.

Offset Address: 0x0810 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:9]	-	reserved	Reserved	0x0000000
[8]	RW	cil0_rst_req	CIL 0 reset 0: reset deasserted 1: reset	0x0
[7:1]	-	reserved	Reserved	0x00
[0]	RW	phycil0_cken	PHY CIL 0 clock gating 0: Disable the clock. 1: Enable the clock.	0x0

## PHYCFG\_MODE

PHYCFG\_MOD is the PHY configuration mode register.

Offset Address: 0x0818 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	Reserved	0x00000000
[6:4]	RW	phycil0_1_cfg_mode	PHY 0 lane 1/3 configuration mode 000: The PHY 0 configuration is controlled by PHYCFG_EN and FSM. (MIPI) 001: The PHY 0 configuration is controlled by PHYCFG_EN. (LVDS) 010: The PHY 0 configuration is directly controlled by the register value. 011: The PHY 0 configuration is controlled by PHYCFG_EN and FSM. The enable status of the clock channel is controlled by PHYCFG_EN. Other values: reserved	0x0
[3]	RW	phycil0_cfg_mode_sel	PHY0 configuration mode 0: controlled by PHYCFG_EN 1: controlled by the register value (CMOS)	0x0
[2:0]	RW	phycil0_0_cfg_mode	PHY 0 lane 0/2 configuration mode 000: The PHY 0 configuration is controlled by PHYCFG_EN and FSM.	0x0



Bits	Access	Name	Description	Reset
			(MIPI) 001: The PHY 0 configuration is controlled by PHYCFG_EN. (LVDS) 010: The PHY 0 configuration is directly controlled by the register value. 011: The PHY 0 configuration is controlled by PHYCFG_EN and FSM. The enable status of the clock channel is controlled by PHYCFG_EN. Other values: reserved	

## PHYCFG\_EN

PHYCFG\_EN is the PHY configuration enable register.

Offset Address: 0x081C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	WO	phycil0_cfg_en	PHY 0 configuration enable register 0: disabled 1: enabled	0x0

## CHNO\_CLR\_EN

CHNO\_CLR\_EN is the channel 0 forcible setting-to-1 enable register.

Offset Address: 0x0824 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	chn0_clr_en_align	Forcible setting-to-1 control for channel 0 ALIGN module 0: disable the setting-to-1 request 1: enable the setting-to-1 request	0x0
[0]	RW	chn0_clr_en_lvds	Forcible setting-to-1 control for channel 0 LVDS_CTRL module 0: disable the setting-to-1 request	0x0



Bits	Access	Name	Description	Reset
			1: enable the setting-to-1 request	

## CHN1\_CLR\_EN

CHN1\_CLR\_EN is the channel 1 forcible setting-to-1 enable register.

Offset Address: 0x082C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	chn1_clr_en_align	Forcible setting-to-1 control for channel 1 ALIGN module 0: disable the setting-to-1 request 1: enable the setting-to-1 request	0x0
[0]	RW	chn1_clr_en_lvds	Forcible setting-to-1 control for channel 1 LVDS_CTRL module 0: disable the setting-to-1 request 1: enable the setting-to-1 request	0x0

## MIPI\_INT\_RAW

MIPI\_INT\_RAW is the MIPI system raw interrupt status register.

Offset Address: 0x0FF0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5]	WC	int_chn1_raw	Channel 1 interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[4]	WC	int_chn0_raw	Channel 0 interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	WC	int_phycil0_raw	PHYCIL 0 raw interrupt status 0: There is no raw interrupt.	0x0



Bits	Access	Name	Description	Reset
			1: There is a raw interrupt.	

## MIPI\_INT\_ST

MIPI\_INT\_ST is the MIPI system interrupt status register.

Offset Address: 0x0FF4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5]	RO	int_chn1_st	Channel 1 interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	int_chn0_st	Channel 0 interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RO	int_phycil0_st	PHYCIL 0 interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## MIPI\_INT\_MSK

MIPI\_INT\_MSK is the MIPI system interrupt mask register.

Offset Address: 0x0FF8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5]	RW	int_chn1_mask	Channel 1 interrupt enable 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[4]	RW	int_chn0_mask	Channel 0 interrupt enable 0: This interrupt is masked. 1: This interrupt is enabled.	0x0



Bits	Access	Name	Description	Reset
[3:1]	-	reserved	Reserved	0x0
[0]	RW	int_phycil0_mask	PHYCIL 0 interrupt enable 0: This interrupt is masked. 1: This interrupt is enabled.	0x0

## MIPI(N)\_CRC\_INTR\_RAW

MIPI(N)\_CRC\_INTR\_RAW is the MIPI (N) CSI raw interrupt status register.

Offset Address: 0x1020+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	Reserved	0x000
[21]	RO	frame_s_e_num_mismatch_vc1_raw	Status of the raw interrupt when the SOF and EOF IDs of virtual channel 1 do not match. Writing <b>1</b> clears this interrupt. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[20]	RO	frame_s_e_num_mismatch_vc0_raw	Status of the raw interrupt when the SOF and EOF IDs of virtual channel 0 do not match. Writing <b>1</b> clears this interrupt. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[19:18]	-	reserved	Reserved	0x0
[17]	RO	frame_num_err_vc1_raw	Status of the raw interrupt when the frame ID of virtual channel 1 has an error. Frame loss may have occurred. Writing <b>1</b> clears this interrupt. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[16]	RO	frame_num_err_vc0_raw	Status of the raw interrupt when the frame ID of virtual channel 0 has an error. Frame loss may have occurred. Writing <b>1</b> clears this interrupt. 0: No interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
			1: An interrupt is generated.	
[15:9]	-	reserved	Reserved	0x00
[8]	RO	ecc_err_mult_raw	Status of the raw interrupt for the packet ECC. At least two errors are detected and cannot be corrected by ECC. Writing <b>1</b> clears this interrupt. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7:6]	-	reserved	Reserved	0x0
[5]	RO	ecc_err_vc1_raw	Status of the raw interrupt for a 1-bit ECC error of virtual channel 1. The error has been corrected by ECC. Writing <b>1</b> clears this interrupt. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	ecc_err_vc0_raw	Status of the raw interrupt for a 1-bit ECC error of virtual channel 0. The error has been corrected by ECC. Writing <b>1</b> clears this interrupt. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RO	crc_err_vc1_raw	Status of the raw interrupt for a data CRC error of virtual channel 1 Writing <b>1</b> clears this interrupt. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	crc_err_vc0_raw	Status of the raw interrupt for a data CRC error of virtual channel 0 Writing <b>1</b> clears this interrupt. 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## MIPI(N)\_CRC\_INTR\_ST

MIPI(N)\_CRC\_INTR\_ST is the MIPI (N) CSI masked interrupt status register.



Offset Address: 0x1024+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:22]	-	reserved	Reserved	0x000
[21]	RO	frame_s_e_num_mismatch_vc1_st	Status of the interrupt generated when the SOF and EOD IDs of virtual channel 1 do not match and the error is handled by the interrupt mask register. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[20]	RO	frame_s_e_num_mismatch_vc0_st	Status of the interrupt generated when the SOF and EOD IDs of virtual channel 0 do not match and the error is handled by the interrupt mask register. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[19:18]	-	reserved	Reserved	0x0
[17]	RO	frame_num_err_vc1_st	Status of the interrupt generated when the frame ID of virtual channel 1 has an error, which is handled by the interrupt mask register. Frame loss may have occurred. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[16]	RO	frame_num_err_vc0_st	Status of the interrupt generated when the frame ID of virtual channel 0 has an error, which is handled by the interrupt mask register. Frame loss may have occurred. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15:9]	-	reserved	Reserved	0x00
[8]	RO	ecc_err_mult_st	Status of the interrupt generated when the packet ECC has at least two ECC errors that cannot be corrected and are handled by the interrupt mask register. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7:6]	-	reserved	Reserved	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[5]	RO	ecc_err_vc1_st	Status of the interrupt generated when virtual channel 1 has a 1-bit ECC error, which has been corrected by ECC and has been handled by the interrupt mask register.  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	ecc_err_vc0_st	Status of the interrupt generated when virtual channel 0 has a 1-bit ECC error, which has been corrected by ECC and has been handled by the interrupt mask register.  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RO	crc_err_vc1_st	Status of the interrupt generated when virtual channel 1 has a data CRC error, which is handled by the interrupt mask register.  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	crc_err_vc0_st	Status of the interrupt generated when virtual channel 0 has a data CRC error, which is handled by the interrupt mask register.  0: No interrupt is generated. 1: An interrupt is generated.	0x0

## **MIPI(N)\_CRC\_INTR\_MSK**

MIPI(N)\_CRC\_INTR\_MSK is the MIPI (N) CSI interrupt mask register.

Offset Address: 0x1028+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:22]	-	reserved	Reserved	0x000
[21]	RW	frame_s_e_num_mismatch_vc1_msk	frame_s_e_num_mismatch_vc1_raw interrupt mask register	0x0



Bits	Access	Name	Description	Reset
			0: This interrupt is masked. 1: This interrupt is enabled.	
[20]	RW	frame_s_e_num_mismatch_vc0_msk	frame_s_e_num_mismatch_vc0_raw interrupt mask register 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[19:18]	-	reserved	Reserved	0x0
[17]	RW	frame_num_err_vc1_msk	frame_num_err_vc1_raw interrupt mask register 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[16]	RW	frame_num_err_vc0_msk	frame_num_err_vc0_raw interrupt mask register 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[15:9]	-	reserved	Reserved	0x00
[8]	RW	ecc_err_mult_msk	ecc_err_mult_raw interrupt mask register 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[7:6]	-	reserved	Reserved	0x0
[5]	RW	ecc_err_vc1_msk	ecc_err_vc1_raw interrupt mask register 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[4]	RW	ecc_err_vc0_msk	ecc_err_vc0_raw interrupt mask register 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RW	crc_err_vc1_msk	crc_err_vc1_raw interrupt mask register 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[0]	RW	crc_err_vc0_msk	crc_err_vc0_raw interrupt mask register 0: This interrupt is masked. 1: This interrupt is enabled.	0x0



## MIPI(N)\_USERDEF\_DT

MIPI(N)\_USERDEF\_DT is the MIPI (N) pixel bit width configuration register for the user-defined data type.

Offset Address: 0x1100+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	Reserved	0x00000000
[6:4]	RW	user_def1_dt	Bit width of the transferred pixels when the data type is user_def1  000: 8bit 001: 10bit 010: 12bit 011: 14bit 100: 16bit  Other values: embed	0x0
[3]	-	reserved	Reserved	0x0
[2:0]	RW	user_def0_dt	Bit width of the transferred pixels when the data type is user_def0  000: 8bit 001: 10bit 010: 12bit 011: 14bit 100: 16bit  Other values: embed	0x0

## MIPI(N)\_USER\_DEF

MIPI(N)\_USER\_DEF is the MIPI (N) configuration enable register for the user-defined data type.

Offset Address: 0x1104+N×0x1000 Total Reset Value: 0x1036\_3534

Bits	Access	Name	Description	Reset
[31:14]	-	reserved	Reserved	0x040D8
[13:8]	RW	user_def1	User-defined data type 1  The value of this field is used for	0x35



			matching the type of the sensor output data. This field works in pairs with user_def1_dt.	
[7:6]	-	reserved	Reserved	0x0
[5:0]	RW	user_def0	User-defined data type 0 The value of this field is used for matching the type of the sensor output data. This field works in pairs with user_def0_dt.	0x34

## MIPI(N)\_CTRL\_MODE\_HS

MIPI(N)\_CTRL\_MODE\_HS is the MIPI (N) operating mode enable register.

Offset Address: 0x1108+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:9]	-	reserved	Reserved	0x0000000
[8]	RW	user_def_en	User-defined mode enable 0: disabled 1: enabled	0x0
[7:5]	-	reserved	Reserved	0x0
[4]	RW	vc_mode	Whether to enable the share Vsync mode when the virtual channel distinguishes data of different exposure lengths 0: No. 1: Yes.	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	hdr_mode	Whether to enable the data type to distinguish data of different exposure lengths 0: No. 1: Yes.	0x0

## MIPI(N)\_DOL\_ID\_CODE0

MIPI(N)\_DOL\_ID\_CODE0 is the MIPI (N) DOL mode frame identification 0 register.



Offset Address: 0x1200+N×0x1000 Total Reset Value: 0x0242\_0241

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RW	id_code_reg1	ID of the $N$ th SEF 1 frame in MIPI DOL mode	0x0242
[15:0]	RW	id_code_reg0	ID of the $N$ th LEF frame in MIPI DOL mode	0x0241

### **MIPI(N)\_DOL\_ID\_CODE1**

MIPI(N)\_DOL\_ID\_CODE1 is the MIPI (N) DOL mode frame identification 1 register.

Offset Address: 0x1204+N×0x1000 Total Reset Value: 0x0251\_0244

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RW	id_code_reg3	ID of the $N+1$ th LEF frame in MIPI DOL mode	0x0251
[15:0]	RW	id_code_reg2	ID of the $N$ th SEF 2 frame in MIPI DOL mode	0x0244

### **MIPI(N)\_DOL\_ID\_CODE2**

MIPI(N)\_DOL\_ID\_CODE2 is the MIPI (N) DOL mode frame identification 2 register.

Offset Address: 0x1208+N×0x1000 Total Reset Value: 0x0254\_0252

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RW	id_code_reg5	ID of the $N+1$ th SEF 2 frame in MIPI DOL mode	0x0254
[15:0]	RW	id_code_reg4	ID of the $N+1$ th SEF 1 frame in MIPI DOL mode	0x0252

### **MIPI(N)\_CROP\_START\_CHN0**

MIPI(N)\_CROP\_START\_CHN0 is the size register for the start point of MIPI (N) channel 0 cropping.

Offset Address: 0x1210+N×0x1000 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_start_y_chn0	Height of the start point for channel 0 cropping	0x0000
[15:0]	RW	mipi_start_x_chn0	Width of the start point for channel 0 cropping	0x0000

## MIPI(N)\_CROP\_START\_CHN1

MIPI(N)\_CROP\_START\_CHN1 is the size register for the start point of MIPI (N) channel 1 cropping.

Offset Address: 0x1214+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_start_y_chn1	Height of the start point for channel 1 cropping	0x0000
[15:0]	RW	mipi_start_x_chn1	Width of the start point for channel 1 cropping	0x0000

## MIPI(N)\_IMGSIZE

MIPI(N)\_IMGSIZE is the MIPI (N) image size register.

Offset Address: 0x1224+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	mipi_imgheight	Image height The configured value is the actual value minus 1.	0x0000
[15:0]	RW	mipi_imgwidth	Image width The configured value is the actual value minus 1.	0x0000

## MIPI(N)\_CTRL\_MODE\_PIXEL

MIPI(N)\_CTRL\_MODE\_PIXEL is the MIPI (N) output operating mode enable register.

Offset Address: 0x1230+N×0x1000 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19]	RW	mipi_yuv_420_nolegacy_en	YUV420 non-legacy enable 0: disabled 1: enabled	0x0
[18]	RW	mipi_yuv_422_en	YUV422 enable 0: disabled 1: enabled	0x0
[17]	RW	mipi_double_yuv_en	Bit width select of YUV output in MIPI mode 0: one pixel per cycle Other values: reserved	0x0
[16]	RW	mipi_double_pix_en	Bit width of the output pixel in MIPI mode 0: one pixel per cycle Other values: reserved	0x0
[15]	-	reserved	Reserved	0x0
[14:13]	RW	stagger_frm_num	Number of frames in staggered HDR mode 01: 2-frame HDR Other values: reserved	0x0
[12]	RW	stagger_hdr_mode	OmniVision staggered HDR mode enable 0: disabled 1: enabled	0x0
[11]	RW	sync_clear_en	EOF auto clear enable 0: disabled 1: enabled	0x0
[10:5]	-	reserved	Reserved	0x00
[4]	RW	mipi_dol_mode	MIPI DOL mode enable 0: disabled 1: enabled	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	crop_en	Cropping enable 0: disabled	0x0



Bits	Access	Name	Description	Reset
			1: enabled	

## MIPI(N)\_DUMMY\_PIX\_REG

MIPI(N)\_DUMMY\_PIX\_REG is the MIPI (N) dummy line pixel value register.

Offset Address: 0x1240+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	dummy_pix_reg	Pixel value of the dummy line in staggered HDR mode	0x0000

## MIPI(N)\_IMGSIZE0\_STATIS

MIPI(N)\_IMGSIZE0\_STATIS is the MIPI (N) VC 0 transferred image size register.

Offset Address: 0x1250+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_statis_vc0	Height of the previous frame of image transferred by virtual channel 2 in MIPI mode	0x0000
[15:0]	RO	imgwidth_statis_vc0	Width of the previous frame of image transferred by virtual channel 1 in MIPI mode	0x0000

## MIPI(N)\_IMGSIZE1\_STATIS

MIPI(N)\_IMGSIZE1\_STATIS is the MIPI (N) VC 1 transferred image size register.

Offset Address: 0x1254+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	imgheight_statis_vc1	Height of the previous frame of image transferred by virtual channel 1 in MIPI mode	0x0000
[15:0]	RO	imgwidth_statis_vc	Width of the previous frame of image	0x0000



		1	transferred by virtual channel 1 in MIPI mode	
--	--	---	---	--

## MIPI(N)\_CTRL\_INT\_RAW

MIPI(N)\_CTRL\_INT\_RAW is the MIPI (N) read data error raw interrupt status register.

Offset Address: 0x12F0+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:18]	-	reserved	Reserved	0x0000
[17]	WC	int_dfifo_rderr_raw	Raw interrupt status of MIPI CTRL read data FIFO 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[16]	WC	int_cfifo_rderr_raw	Raw interrupt status of MIPI CTRL read command FIFO 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[15:5]	-	reserved	Reserved	0x000
[4]	WC	int_vsync_raw	MIPI CTRL Vsync raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	WC	int_dfifo_wrerr_rw	Raw interrupt status of MIPI CTRL write data FIFO 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[0]	WC	int_cfifo_wrerr_rw	Raw interrupt status of MIPI CTRL write command FIFO 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0

## MIPI(N)\_CTRL\_INT

MIPI(N)\_CTRL\_INT is the MIPI (N) read data error interrupt status register.



Offset Address: 0x12F4+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:18]	-	reserved	Reserved	0x0000
[17]	RO	int_dfifo_rderr_st	Interrupt status of MIPI CTRL read data FIFO  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[16]	RO	int_cfifo_rderr_st	Interrupt status of MIPI CTRL read command FIFO  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15:5]	-	reserved	Reserved	0x000
[4]	RO	int_vsync_st	MIPI CTRL Vsync interrupt status  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RO	int_dfifo_wrerr_st	Interrupt status of MIPI CTRL write data FIFO  0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	int_cfifo_wrerr_st	Interrupt status of MIPI CTRL write command FIFO  0: No interrupt is generated. 1: An interrupt is generated.	0x0

## MIPI(N)\_CTRL\_INT\_MSK

MIPI(N)\_CTRL\_INT\_MSK is the MIPI (N) read data error interrupt mask register.

Offset Address: 0x12F8+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:18]	-	reserved	Reserved	0x0000
[17]	RW	int_dfifo_rderr_ms_k	Interrupt enable for MIPI CTRL read data FIFO  0: This interrupt is masked.	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
			1: This interrupt is enabled.	
[16]	RW	int_cfifo_rderr_ms_k	Interrupt enable for MIPI CTRL read command FIFO 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[15:5]	-	reserved	Reserved	0x000
[4]	RW	int_vsync_msk	Interrupt enable for MIPI CTR Vsync 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RW	int_dfifo_wrerr_ms_k	Interrupt enable for MIPI CTRL write data FIFO 0: This interrupt is masked. 1: This interrupt is enabled.	0x0
[0]	RW	int_cfifo_wrerr_ms_k	Interrupt enable for MIPI CTRL write command FIFO 0: This interrupt is masked. 1: This interrupt is enabled.	0x0

## LVDS(N)\_WDR

LVDS(N)\_WDR is the LVDS (N) WDR control register.

Offset Address: 0x1300+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	-	reserved	Reserved	0x0000
[15:12]	RW	lvds_wdr_id_shift	IMX136 frame ID shift register, indicating the number of bits to be shifted rightwards to make the frame ID be in bit 0	0x0
[11:8]	RW	lvds_wdr_mode	LVDS WDR mode select 0x0: WDR mode with the SOF-EOF flag. In this mode, the long and short exposure frames have independent sync codes.	0x0



Bits	Access	Name	Description	Reset
			<p>0x2: HiSPi WDR mode. In this mode, the long and short exposure frames share one SOF-EOF flag, and the first several lines of the short exposure frame are stuffed with 0x04.</p> <p>0x4: SONY DOL WDR mode. In this mode, the sync code consists of four fields, the SAV-EAV flag is used, and the long and short exposure frames have independent sync codes.</p> <p>0x5: SONY DOL WDR mode. In this mode, the sync code consists of four fields, the SAV-EAV flag is used, the long and short exposure frames share one group of sync codes, and there are blanking regions between the long exposure data and short exposure data.</p> <p>0x6: SONY DOL WDR mode. In this mode, the sync code consists of five fields. The fifth field indicates whether the frame is a long frame or a short frame.</p> <p>0x8: SONY IMX136 frame WDR mode. Do not enable WDR_EN in this mode.</p> <p>0x9: Sharp long/short exposure 2-line alternating mode. The long frame is before the short frame.</p> <p>0xA: Sharp long/short exposure 2-line alternating mode. The long frame is after the short frame.</p> <p>Other values: reserved</p>	
[7:6]	-	reserved	Reserved	0x0
[5:4]	RW	lvds_wdr_num	<p>WDR mode configuration</p> <p>00: reserved</p> <p>01: 2-frame WDR</p> <p>Other values: reserved</p>	0x0
[3:1]	-	reserved	Reserved	0x0
[0]	RW	lvds_wdr_en	<p>WDR enable</p> <p>0: linear mode</p> <p>1: WDR mode</p>	0x0



## LVDS(N)\_DOLSCD\_HBLK

LVDS(N)\_DOLSCD\_HBLK is the LVDS (N) SCD control register.

Offset Address: 0x1304+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	dol_hblank2	HBLANK 2 value in DOL mode	0x0000
[15:0]	RW	dol_hblank1	HBLANK 1 value in DOL mode	0x0000

## LVDS(N)\_CTRL

LVDS(N)\_CTRL is the LVDS (N) control register.

Offset Address: 0x1308+N×0x1000 Total Reset Value: 0x0000\_0330

Bits	Access	Name	Description	Reset
[31:19]	-	reserved	Reserved	0x0000
[18:16]	RW	lvds_split_mode	LVDS/HiSPi sync code transfer type 000: per lane mode Other values: reserved	0x0
[15:13]	-	reserved	Reserved	0x0
[12]	RW	lvds_crop_en	LVDS/HiSPi image cropping enable 0: disabled 1: enabled	0x0
[11:10]	-	reserved	Reserved	0x0
[9]	RW	lvds_code_big_endian	Serial bit transmission sequence for sync_code of raw data to be transmitted in LVDS or HiSPi mode 0: LSBs are transmitted first. The serial data sequence of sync_code received is bit 0, bit 1... bit 11. 1: MSBs are transmitted first. The serial data sequence of the sync_code received is bit 11, bit 10... bit 0.	0x1
[8]	RW	lvds_pix_big_endian	Serial bit transmission sequence for the valid pixels of raw data to be	0x1



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
			transmitted in LVDS or HiSPi mode 0: LSBs are transmitted first. The serial data sequence of the valid pixels received is bit 0, bit 1... bit 11. 1: MSBs are transmitted first. The serial data sequence of the valid pixels received is bit 11, bit 10... bit 0.	
[7]	-	reserved	Reserved	0x0
[6:4]	RW	lvds_raw_type	Type of raw data to be transmitted in LVDS or HiSPi mode 001: 8-bit raw data 010: 10-bit raw data 011: 12-bit raw data 100: 14-bit raw data Other values: reserved	0x3
[3:1]	-	reserved	Reserved	0x0
[0]	RW	lvds_sync_mode	Frame/Line sync mode in LVDS mode 0: SOF/EOF/SOL/EOL sync mode. SOF marks the start of the first line of the valid region. EOF marks the end of the last line in the valid region. SOL and EOL mark the start and end of other valid regions. 1: SAV/EAV sync mode. SAV (invalid) and EAV (invalid) mark the invalid data of the blanking region. SAV (valid) and EAV (valid) mark the pixel data of the valid region.	0x0

## LVDS(N)\_IMGSIZE

LVDS(N)\_IMGSIZE is the LVDS (N) image size register.

Offset Address: 0x130C+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RW	lvds_imgheight	Image height The configured value is the actual value minus 1.	0x0000



[15:0]	RW	lvds_imgwidth_lane	Width of the image transferred by each channel The configured value is the actual value minus 1.	0x0000
--------	----	--------------------	---	--------

## LVDS(N)\_CROP\_START0

LVDS(N)\_CROP\_START0 is the cropping register for the first LVDS (N) frame.

Offset Address: 0x1310+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lvds_start_y0	Height of the cropping start point for the first frame	0x0000
[15:0]	RW	lvds_start_x0_lane	Width of the cropping start point for the first frame The configured value is the start pixel to be cropped divided by the number of channels.	0x0000

## LVDS(N)\_CROP\_START1

LVDS(N)\_CROP\_START1 is the cropping register for the second LVDS (N) frame.

Offset Address: 0x1314+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lvds_start_y1	Height of the cropping start point for the second frame	0x0000
[15:0]	RW	lvds_start_x1_lane	Width of the cropping start point for the second frame The configured value is the start pixel to be cropped divided by the number of channels.	0x0000

## LVDS(N)\_LANE(M)\_SOF\_01

LVDS(N)\_LANE(M)\_SOF\_01 is the SOF sync code configuration register for lane (M) in LVDS or HiSPI mode.



Offset Address: 0x1320+M×0x20+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(m)_sof_1	SOF sync code of frame 1 for lane (M) in LVDS or HiSPi mode This field is configurable only in WDR mode, and the configuration is invalid in linear mode.	0x0000
[15:0]	RW	lane(m)_sof_0	SOF sync code of frame 0 for lane (M) in LVDS or HiSPi mode In WDR mode, the value is the SOF of frame 0. In linear mode, the value is the SOF.	0x0000

### LVDS(N)\_LANE(M)\_EOF\_01

LVDS(N)\_LANE(M)\_EOF\_01 EOF sync code configuration register for lane (M) in LVDS or HiSPi mode.

Offset Address: 0x1328+M×0x20+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(m)_eof_1	EOF sync code of frame 1 in LVDS or HiSPi mode, lane (M) EOF of frame 1. This field is configurable only in WDR mode, and the configuration is invalid in linear mode.	0x0000
[15:0]	RW	lane(m)_eof_0	EOF sync code of frame 0 in LVDS or HiSPi mode, lane (M) EOF of frame 0. In WDR mode, the value is the EOF of frame 0. In linear mode, the value is the EOF.	0x0000

### LVDS(N)\_LANE(M)\_SOL\_01

LVDS(N)\_LANE(M)\_SOL\_01 is the SOL sync code configuration register for lane (M) in LVDS or HiSPi mode.

Offset Address: 0x1330+M×0x20+N×0x1000 Total Reset Value: 0x0000\_0000



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RW	lane(m)_sol_1	SOL sync code of frame 1 in LVDS or HiSPI mode, lane (M) SOL of frame 1 This field is configurable only in WDR mode, and the configuration is invalid in linear mode.	0x0000
[15:0]	RW	lane(m)_sol_0	SOL sync code of frame 1 in LVDS or HiSPI mode, lane (M) SOL of frame 0 In WDR mode, the value is the SOL of frame 0. In linear mode, the value is the SOL.	0x0000

### LVDS(N)\_LANE(M)\_EOL\_01

LVDS(N)\_LANE(M)\_EOL\_01 is the EOL sync code configuration register for lane (M) in LVDS or HiSPI mode.

Offset Address: 0x1338+M×0x20+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RW	lane(m)_eol_1	EOL sync code of frame 1 in LVDS or HiSPI mode, lane (M) EOL of frame 1 This field is configurable only in WDR mode, and the configuration is invalid in linear mode.	0x0000
[15:0]	RW	lane(m)_eol_0	EOL sync code of frame 0 in LVDS or HiSPI mode, lane (M) EOL of frame 0 In WDR mode, the value is the EOL of frame 0. In linear mode, the value is the EOL.	0x0000

### LVDS(N)\_LANE(M)\_NXT\_SOFR\_01

LVDS(N)\_LANE(M)\_NXT\_SOFR\_01 is the lane (M) SOFR sync code configuration register for frame (N+ 1) in LVDS or HiSPI mode

Offset Address: 0x1520+M×0x20+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	RW	lane(m)_nxt_sof_1	SOFR sync code of frame 1 in LVDS or	0x0000



Bits	Access	Name	Description	Reset
			HiSPi mode, lane (M) SOF of frame 1 This field is configurable only in WDR mode, and the configuration is invalid in linear mode.	
[15:0]	RW	lane(m)_nxt_sof_0	SOF sync code of frame 0 in LVDS or HiSPi mode, lane (M) SOF of frame 0 In WDR mode, the value is the SOF of frame 0. In linear mode, the value is the SOF.	0x0000

### LVDS(N)\_LANE(M)\_NXT\_EOF\_01

LVDS(N)\_LANE(M)\_NXT\_EOF\_01 is the lane (M) EOF sync code configuration register for frame ( $N+1$ ) in LVDS or HiSPi mode.

Offset Address: 0x1528+M×0x20+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(m)_nxt_eof_1	EOF sync code of frame 1 in LVDS or HiSPi mode, lane (M) EOF of frame 1 This field is configurable only in WDR mode, and the configuration is invalid in linear mode.	0x0000
[15:0]	RW	lane(m)_nxt_eof_0	EOF sync code of frame 0 in LVDS or HiSPi mode, lane (M) EOF of frame 0 In WDR mode, the value is the EOF of frame 0. In linear mode, the value is the EOF.	0x0000

### LVDS(N)\_LANE(M)\_NXT\_SOL\_01

LVDS(N)\_LANE(M)\_NXT\_SOL\_01 is the lane (M) SOL sync code configuration register for frame ( $N+1$ ) in LVDS or HiSPi mode.

Offset Address: 0x1530+M×0x20+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(m)_nxt_sol_1	SOL sync code of frame 1 in LVDS or HiSPi mode, lane (M) SOL of frame 1 This field is configurable only in WDR mode.	0x0000



			mode, and the configuration is invalid in linear mode.	
[15:0]	RW	lane(m)_nxt_sol_0	SOL sync code of frame 0 in LVDS or HiSPi mode, lane (M) SOL of frame 0 In WDR mode, the value is the SOL of frame 0. In linear mode, the value is the SOL.	0x0000

### LVDS(N)\_LANE(M)\_NXT\_EOL\_01

LVDS(N)\_LANE(M)\_NXT\_EOL\_01 is the lane (M) EOL sync code configuration register for frame ( $N + 1$ ) in LVDS or HiSPi mode.

Offset Address: 0x1538+M×0x20+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	lane(m)_nxt_eol_1	EOL sync code of frame 1 in LVDS or HiSPi mode, lane (M) EOL of frame 1 This field is configurable only in WDR mode, and the configuration is invalid in linear mode.	0x0000
[15:0]	RW	lane(m)_nxt_eol_0	EOL sync code of frame 0 in LVDS or HiSPi mode, lane (M) EOL of frame 0 In WDR mode, the value is the EOL of frame 0. In linear mode, the value is the EOL.	0x0000

### LVDS(N)\_LI\_WORD0

LVDS(N)\_LI\_WORD0 is the LVDS DOL mode frame 0 LI register.

Offset Address: 0x1720+N×0x1000 Total Reset Value: 0x0211\_0201

Bits	Access	Name	Description	Reset
[31:16]	RW	li_word0_1	LEF line information in DOL mode (for frame $N + 1$ )	0x0211
[15:0]	RW	li_word0_0	LEF line information in DOL mode (for frame $N$ )	0x0201



## LVDS(N)\_LI\_WORD1

LVDS(N)\_LI\_WORD1 is the LVDS DOL mode frame 1 LI register.

Offset Address: 0x1724+N×0x1000 Total Reset Value: 0x0212\_0202

Bits	Access	Name	Description	Reset
[31:16]	RW	li_word1_1	SEF 1 line information in DOL mode (for frame N + 1)	0x0212
[15:0]	RW	li_word1_0	SEF 1 line information in DOL mode (for frame N)	0x0202

## LVDS(N)\_IMGSIZE0\_STATIS

LVDS(N)\_IMGSIZE0\_STATIS is the LVDS LEF image size statistics register.

Offset Address: 0x1780+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	lvds_imgheight0	Height of the image transferred by virtual channel 0 in LVDS or HiSPi mode	0x0000
[15:0]	RO	lvds_imgwidth0	Width of the image transferred by virtual channel 0 in LVDS or HiSPi mode	0x0000

## LVDS(N)\_IMGSIZE1\_STATIS

LVDS(N)\_IMGSIZE1\_STATIS is the LVDS SEF 1 image size statistics register.

Offset Address: 0x1784+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	lvds_imgheight1	Height of the image transferred by virtual channel 1 in LVDS or HiSPi mode	0x0000
[15:0]	RO	lvds_imgwidth1	Width of the image transferred by virtual channel 1 in LVDS or HiSPi mode	0x0000

## LVDS(N)\_OUTPUT\_PIX\_NUM

LVDS(N)\_OUTPUT\_PIX\_NUM is the LVDS output pixel bit width selection register

Offset Address: 0x17A0+N×0x1000 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	lvds_double_pix_en	Output pixel bit width selection in LVDS mode 0: one pixel per cycle Other values: reserved	0x0

## LVDS(N)\_CTRL\_INT\_RAW

LVDS(N)\_CTRL\_INT\_RAW is the LVDS read data raw interrupt status register.

Offset Address: 0x17F0+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x0
[28]	WC	lvds_vsync_raw	LVDS Vsync raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[27:25]	-	reserved	Reserved	0x0
[24]	WC	lvds_state_err_raw	LVDS lane sync error raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[23:22]	-	reserved	Reserved	0x0
[21]	WC	link1_rd_err_raw	Link 1 FIFO read error raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[20]	WC	link0_rd_err_raw	Link 0 FIFO read error raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[19:18]	-	reserved	Reserved	0x0
[17]	WC	link1_wr_err_raw	Link 1 FIFO write error raw interrupt status 0: There is no raw interrupt.	0x0



Bits	Access	Name	Description	Reset
			1: There is a raw interrupt.	
[16]	WC	link0_wr_err_raw	Link 0 FIFO write error raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[15:4]	-	reserved	Reserved	0x000
[3]	WC	lane3_sync_err_ra w	Lane 3 sync error raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[2]	WC	lane2_sync_err_ra w	Lane 2 sync error raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[1]	WC	lane1_sync_err_ra w	Lane 1 sync error raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[0]	WC	lane0_sync_err_ra w	Lane 0 sync error raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0

## LVDS(N)\_CTRL\_INT

LVDS(N)\_CTRL\_INT is the LVDS read data interrupt status register.

Offset Address: 0x17F4+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x0
[28]	RO	lvds_vsync_st	LVDS Vsync interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[27:25]	-	reserved	Reserved	0x0
[24]	RO	lvds_state_err_st	LVDS lane sync error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
[23:22]	-	reserved	Reserved	0x0
[21]	RO	link1_rd_err_st	Link 1 FIFO read error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[20]	RO	link0_rd_err_st	Link 0 FIFO read error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[19:18]	-	reserved	Reserved	0x0
[17]	RO	link1_wr_err_st	Link 1 FIFO write error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[16]	RO	link0_wr_err_st	Link 0 FIFO write error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15:4]	-	reserved	Reserved	0x000
[3]	RO	lane3_sync_err_st	Lane 3 sync error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	RO	lane2_sync_err_st	Lane 2 sync error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	lane1_sync_err_st	Lane 1 sync error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	lane0_sync_err_st	Lane 0 sync error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## LVDS(N)\_CTRL\_INT\_MSK

LVDS(N)\_CTRL\_INT\_MSK is the LVDS read data interrupt mask register.

Offset Address: 0x17F8+N×0x1000 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x0
[28]	RW	lvds_vsync_msk	LVDS Vsync interrupt enable 0: disabled 1: enabled	0x0
[27:25]	-	reserved	Reserved	0x0
[24]	RW	lvds_state_err_msk	LVDS lane sync error interrupt enable 0: disabled 1: enabled	0x0
[23:22]	-	reserved	Reserved	0x0
[21]	RW	link1_rd_err_msk	Link 1 FIFO read error interrupt enable 0: disabled 1: enabled	0x0
[20]	RW	link0_rd_err_msk	Link 0 FIFO read error interrupt enable 0: disabled 1: enabled	0x0
[19:18]	-	reserved	Reserved	0x0
[17]	RW	link1_wr_err_msk	Link 1 FIFO write error interrupt enable 0: disabled 1: enabled	0x0
[16]	RW	link0_wr_err_msk	Link 0 FIFO write error interrupt enable 0: disabled 1: enabled	0x0
[15:4]	-	reserved	Reserved	0x0
[3]	RW	lane3_sync_err_ms k	Lane 3 sync error interrupt enable 0: disabled 1: enabled	0x0
[2]	RW	lane2_sync_err_ms k	Lane 2 sync error interrupt enable 0: disabled 1: enabled	0x0
[1]	RW	lane1_sync_err_ms k	Lane 1 sync error interrupt enable 0: disabled 1: enabled	0x0



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[0]	RW	lane0_sync_err_ms_k	Lane 0 sync error interrupt enable 0: disabled 1: enabled	0x0

## LANE\_ID0\_CHN(N)

LANE\_ID0\_CHN(N) is lane priority configuration register 0.

Offset Address: 0x1800+N×0x1000 Total Reset Value: 0x0000\_3210

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:16]	-	reserved	Reserved	0x0000
[15:12]	RW	lane3_id	Lane 3 channel ID, ranging from 0 to 3 The value indicates the actual ID of the channel connected to the image sensor of lane 3.	0x3
[11:8]	RW	lane2_id	Lane 2 channel ID, ranging from 0 to 3 The value indicates the actual ID of the channel connected to the image sensor of lane 2.	0x2
[7:4]	RW	lane1_id	Lane 1 channel ID, ranging from 0 to 3 The value indicates the actual ID of the channel connected to the image sensor of lane 1.	0x1
[3:0]	RW	lane0_id	Lane 0 channel ID, ranging from 0 to 3 The value indicates the actual ID of the channel connected to the image sensor of lane 0.	0x0

## ALIGN(N)\_INT\_RAW

ALIGN(N)\_INT\_RAW is the MIPI\_ALIGN raw interrupt status register.

Offset Address: 0x18F0+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:5]	-	reserved	Reserved	0x00000000



Bits	Access	Name	Description	Reset
[4]	WC	err_lane3_raw	ALIGN lane 3 raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[3]	WC	err_lane2_raw	ALIGN lane 2 raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[2]	WC	err_lane1_raw	ALIGN lane 1 raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[1]	WC	err_lane0_raw	ALIGN lane 0 raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[0]	WC	err_full_raw	ALIGN FIFO raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0

## ALIGN(N)\_INT

ALIGN(N)\_INT is the MIPI\_ALIGN interrupt status register.

Offset Address: 0x18F4+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RO	err_lane3_st	ALIGN lane 3 interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	RO	err_lane2_st	ALIGN lane 2 interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	RO	err_lane1_st	ALIGN lane 1 interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	err_lane0_st	ALIGN lane 0 interrupt status	0x0



Bits	Access	Name	Description	Reset
			0: No interrupt is generated. 1: An interrupt is generated.	
[0]	RO	err_full_st	ALIGN FIFO interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## ALIGN(N)\_INT\_MSK

ALIGN(N)\_INT\_MSK is the MIPI\_ALIGN interrupt mask register.

Offset Address: 0x18F8+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4]	RW	err_lane3_mask	ALIGN lane 3 interrupt enable 0: disabled 1: enabled	0x0
[3]	RW	err_lane2_mask	ALIGN lane 2 interrupt enable 0: disabled 1: enabled	0x0
[2]	RW	err_lane1_mask	ALIGN lane 1 interrupt enable 0: disabled 1: enabled	0x0
[1]	RW	err_lane0_mask	ALIGN lane 0 interrupt enable 0: disabled 1: enabled	0x0
[0]	RW	err_full_mask	ALIGN FIFO interrupt enable 0: disabled 1: enabled	0x0

## CHN(N)\_INT\_RAW

CHN(N)\_INT\_RAW is the channel (N) raw interrupt status register.

Offset Address: 0x1FF0+N×0x1000 Total Reset Value: 0x0000\_0000



<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:4]	-	reserved	Reserved	0x00000000
[3]	WC	int_data_align_raw	DATA_ALIGN raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[2]	WC	int_mipi_ctrl_raw	MIPI_CTRL raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[1]	WC	int_mipi_csi_raw	MIPI_CSI raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0
[0]	WC	int_lvds_ctrl_raw	LVDS_CTRL raw interrupt status 0: There is no raw interrupt. 1: There is a raw interrupt.	0x0

## CHN(N)\_INT

CHN(N)\_INT is the channel (N) interrupt status register.

Offset Address: 0x1FF4+N×0x1000 Total Reset Value: 0x0000\_0000

<b>Bits</b>	<b>Access</b>	<b>Name</b>	<b>Description</b>	<b>Reset</b>
[31:4]	-	reserved	Reserved	0x00000000
[3]	RO	int_data_align_st	DATA_ALIGN interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	RO	int_mipi_ctrl_st	MIPI_CTRL interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	int_mipi_csi_st	MIPI_CSI interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	int_lvds_ctrl_st	LVDS_CTRL interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0



## CHN(N)\_INT\_MASK

CHN(N)\_INT\_MASK is the channel (N) interrupt mask register.

Offset Address: 0x1FF8+N×0x1000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3]	RW	int_data_align_mask	DATA_ALIGN interrupt enable 0: disabled 1: enabled	0x0
[2]	RW	int_mipi_ctrl_mask	MIPI_CTRL interrupt enable 0: disabled 1: enabled	0x0
[1]	RW	int_mipi_csi_mask	MIPI_CSI interrupt enable 0: disabled 1: enabled	0x0
[0]	RW	int_lvds_ctrl_mask	LVDS_CTRL interrupt enable 0: disabled 1: enabled	0x0



## Contents

<b>10 ISP .....</b>	<b>10-1</b>
10.1 Introduction.....	10-1
10.2 Overview.....	10-2
10.3 Operating Mode.....	10-4
10.4 Interrupt System.....	10-5
10.5 Module Functions .....	10-7



## Figures

<b>Figure 10-1</b> Overall structure of the ISP .....	10-2
<b>Figure 10-2</b> ISP_FE structure.....	10-3
<b>Figure 10-3</b> Overall structure of ISP_BE .....	10-3
<b>Figure 10-4</b> Interrupt timing.....	10-6
<b>Figure 10-5</b> Relationship between the valid image area and the horizontal/vertical blanking region .....	10-7
<b>Figure 10-6</b> Color space of valid AWB pixel .....	10-11
<b>Figure 10-7</b> Gamma curve.....	10-13
<b>Figure 10-8</b> Overshoot and undershoot occurrence during sharpening .....	10-14



## Tables

<b>Table 10-1</b> Key parameters in external ISP mode .....	10-4
<b>Table 10-2</b> Key parameters of the position-adjustable modules.....	10-5
<b>Table 10-3</b> Interrupt indicator register .....	10-5
<b>Table 10-4</b> Mapping between the interrupt ID and the interrupt event .....	10-7
<b>Table 10-5</b> AE zonal statistics .....	10-9
<b>Table 10-6</b> AE AE histogram statistics .....	10-10
<b>Table 10-7</b> AWB zonal statistics for mode A.....	10-11



# 10 ISP

## 10.1 Introduction

The image signal processor (ISP) module supports standard sensor image data processing. The ISP module provides basic functions such as automatic white balance (AWB), automatic exposure (AE), demosaic, defect pixel correction (DPC), and lens shading correction (LSC) as well as advanced functions such as wide dynamic range (WDR), dynamic range compression (DRC), and denoising. The ISP mainly supports the following image processing functions:

- Black level correction (BLC)
- Dynamic DPC and defect pixel cluster correction
- Bayer denoising
- Fixed pattern noise (FPN) removal
- Demosaic
- Chromatic aberration correction (CAC)
- Gamma correction
- DRC
- Sensor built-in WDR
- 2F-WDR frame mode
- AWB
- AE
- 3A (AE, AF, AWB) statistics output
- Lens shading correction
- Image sharpening
- DIS
- Automatic anti-fog
- Local contrast enhancement



- Color adaptation (CA)
- 3D noise reduction

The processing capability of the ISP module is as follows:

- Maximum 14-bit Bayer data input in normal mode
- Maximum 14-bit Bayer data input in built-in WDR mode
- Maximum image resolution of 3840 x 2160
- Minimum image resolution of 120 x 120
- Minimum horizontal blanking region of 128 pixels
- Minimum vertical blanking region of 40 lines
- Maximum performance:
  - Hi3516CV610-20B/20S/00B/00S/00G/20G: 3840 x 2160 @ 20 fps or 3200 x 1800 @ 30 fps
  - Hi3516CV610-10B: 2880 x 1620 @ 30 fps

## 10.2 Overview

### Functional Block Diagram

[Figure 10-1](#), [Figure 10-2](#), and [Figure 10-3](#) show the functional block diagram of the ISP module. ISP\_FE mentioned in this document refers to the part before FPN (excluding FPN) in the ISP pipeline, and ISP\_BE refers to the part after FPN (including FPN) in the ISP pipeline.

#### NOTE

In this document,  $U^*$  and  $S^*$  represent the unsigned number and the signed number respectively. For example, U8.8 indicates that the data is an unsigned number consisting of an 8-bit integral part and an 8-bit decimal part. S8.8 indicates that the data is a signed number consisting of an 8-bit integral part (including a sign bit) and an 8-bit decimal part.

**Figure 10-1** Overall structure of the ISP

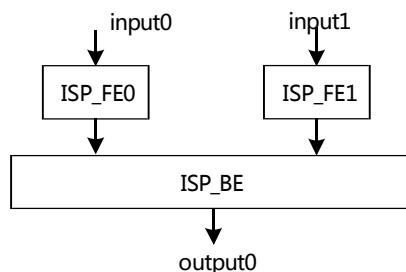




Figure 10-2 ISP\_FE structure

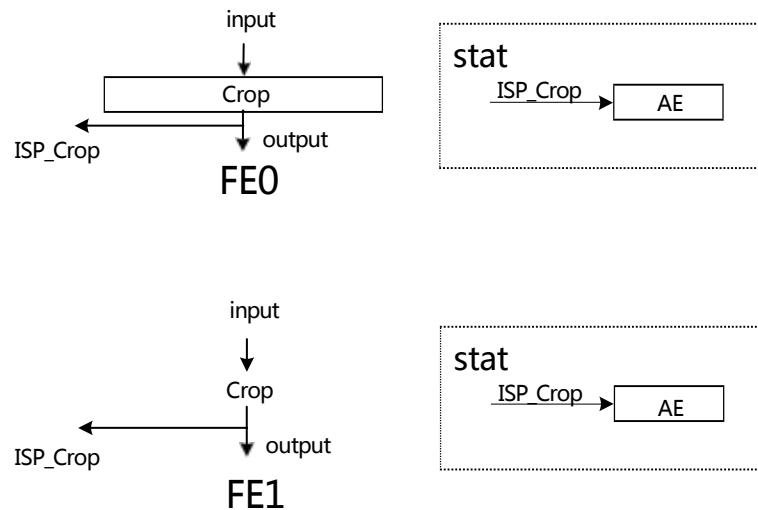
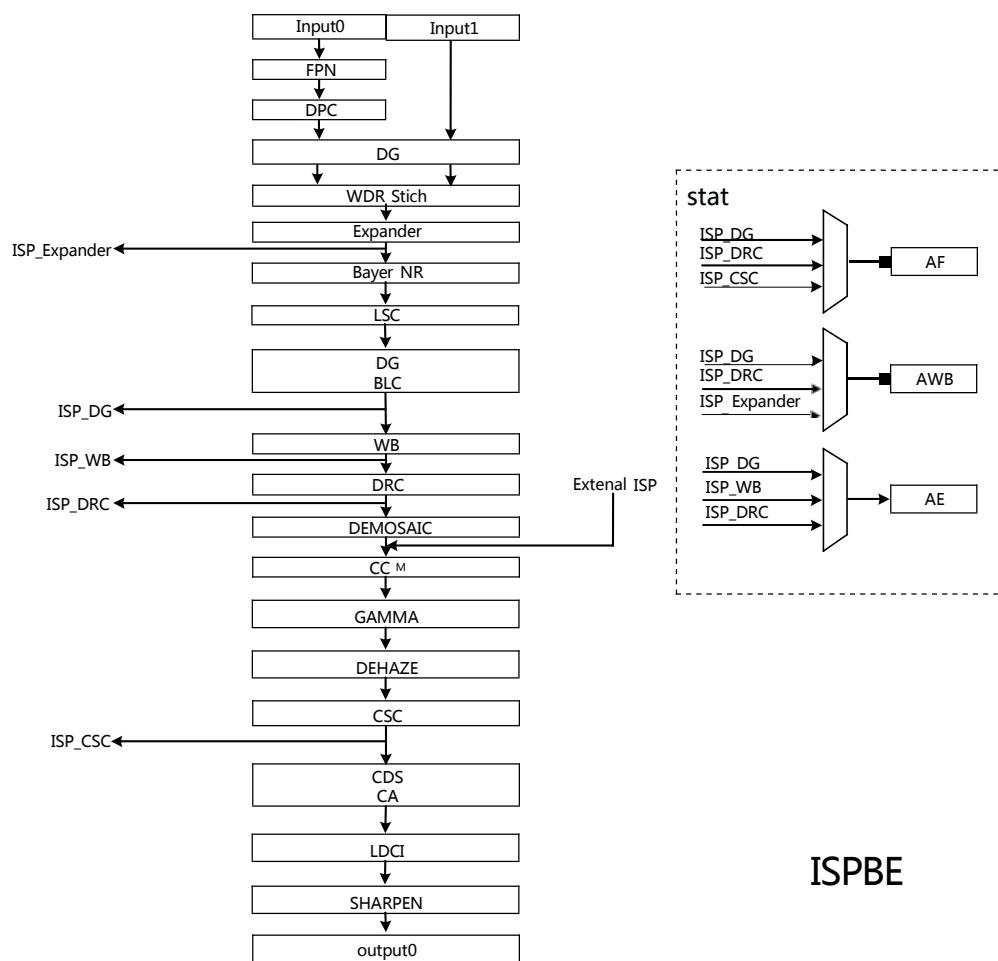


Figure 10-3 Overall structure of ISP\_BE





## 10.3 Operating Mode

The ISP module supports the following operating modes:

- Maximum 14-bit Bayer data input  
When the input data width is less than 14 bits, the upper bits need to be aligned and the lower bits need to be padded with 0s. In this mode, any sequence of the R, Gr, Gb, and B component is supported.
- Sensor built-in WDR  
This mode supports maximum 14 bits Bayer data input.
- 2F-WDR frame mode
- Luminance single-component mode  
The C (U and V) component can be discarded, so that only the Y component is output.
- Time division multiplexing (TDM) for ISP BE

The examples of the TDM application scenarios are as follows:

Example 1: For two ISP\_FE inputs, the clock frequency of ISP\_FE0 and ISP\_FE1 is 300 MHz, respectively. ISP\_BE can process data from ISP\_FE0 and ISP\_FE1 in TDM mode.

### NOTE

The frequencies of ISP\_FE and ISP\_BE can be separately configured. For details, see section 3.2.5 "Register Description" in the *Data Sheet*.

- External ISP mode

[Table 10-1](#) describes the key parameters in external ISP mode.

**Table 10-1** Key parameters in external ISP mode

Parameter	Description
ISP_BE_INPUT_MUX [16]	ISP data input select 0: raw data 1: YUV data

- Adjustable module position

The positions of the DPC, AE, AWB, and AF modules are adjustable.



**Table 10-2** Key parameters of the position-adjustable modules

Parameter	Description
ISP_BE_MODULE_POS[8]	DPC position adjustment register 0: The DCG module is before the Demosaic module. 1: The DCG module is before the WDR module.
ISP_BE_MODULE_POS[5:4]	AF position adjustment register 00: The AF module is after the DG module. 01: The AF module is after the DRC module. 10: The AF module is after the CSC module.
ISP_BE_MODULE_POS[3:2]	AWB position adjustment register 00: The AWB module is after the DG module. 01: The AWB module is after the Expander module. 10: The AWB module is after the DRC module.
ISP_BE_MODULE_POS[1:0]	AE position adjustment register 00: The AE module is after the DG module. 01: The AE module is after the WB module. 10: The AE module is after the DRC module.

## 10.4 Interrupt System

### Function Description

The ISP module has 20 hardware interrupt events. For details, see [Table 10-3](#)

**Table 10-3** Interrupt indicator register

Offset Address	Bit	Description
0x3200F0+FE_N*0x2000	bit[4]	AE statistics completion interrupt
	bit[3]	Configurable ISP FE trigger position interrupt. Any trigger position in the unit of line in the input image valid region is supported.

Offset Address	Bit	Description
	bit[2]	ISP FE register configuration loss interrupt
	bit[1]	ISP FE register update interrupt
	bit[0]	ISP FE frame start interrupt
0x00310	bit[26]	LA statistics completion interrupt
	bit[25]	DEHAZE statistics completion interrupt
	bit[24]	MEM error interrupt
	bit[22]	AF statistics completion interrupt
	bit[21]	AWB statistics completion interrupt
	bit[20]	AE statistics completion interrupt
	bit[18]	ISP BE register configuration loss interrupt
	bit[17]	ISP BE register update interrupt
	bit[16]	ISP BE frame start interrupt

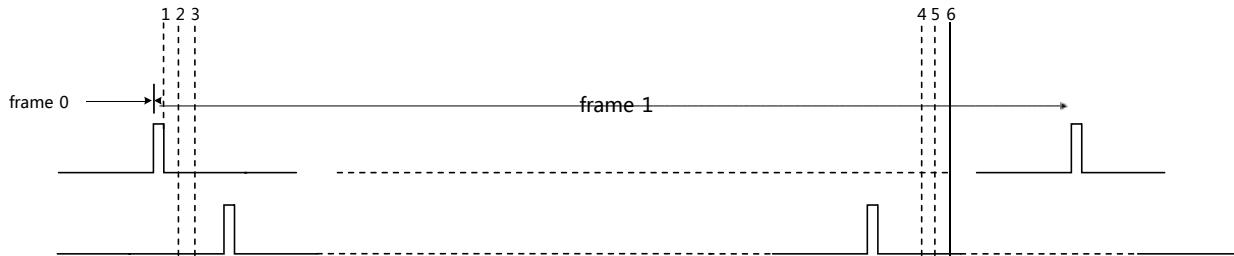
### NOTE

FE\_N corresponds to the FE in the overall ISP architecture. The base address of the FE configuration registers is 0x1100\_0000.

## Interrupt Timing

The positions of many interrupts are determined by the ISP module switch and register configuration. [Figure 10-4](#) shows the position of each interrupt. Note that the interrupt position is arranged based on the time sequence per frame. [Table 10-4](#) describes the mapping between the interrupt ID and the interrupt event shown in [Figure 10-4](#).

**Figure 10-4** Interrupt timing



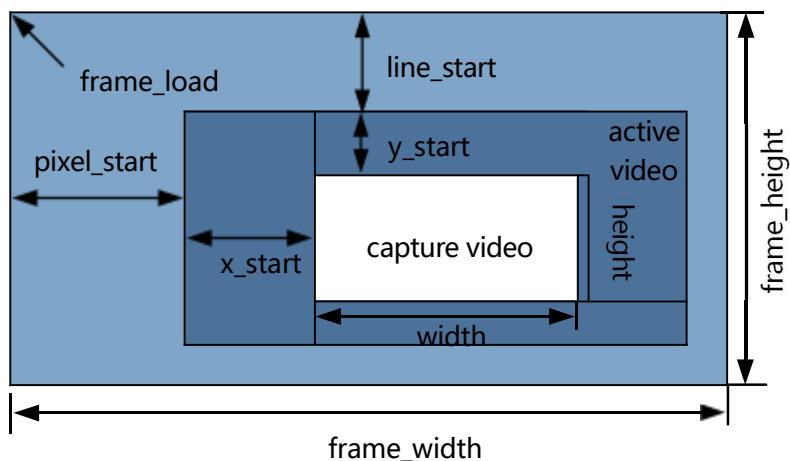
**Table 10-4** Mapping between the interrupt ID and the interrupt event

No.	Interrupt Event
1	fstart(ISP_FE)/fstart(ISP_BE)
2	cfg_loss(ISP_FE)/cfg_loss(ISP_BE) update_cfg(ISP_FE) /update_cfg(ISP_BE)
3	fstart_delay(ISP_FE)
4	ae_int
5	af_int
6	Dehaze statistics completion interrupt LDCI statistics completion interrupt

## 10.5 Module Functions

### Crop

This module crops an input image. The displayed view area is usually covered by the valid video range, and there are some pixels reduced at the relatively valid video boundary as shown in [Figure 10-5](#).

**Figure 10-5** Relationship between the valid image area and the horizontal/vertical blanking region



## FPN

The FPN module corrects the sensor input image based on the calibrated black frame to remove the FPNs of the sensor.

The FPN module supports calibration and correction in frame mode. This module needs to be enabled if the FPNs of the connected sensor are obvious, and does not need to be enabled if the FPNs of the connected sensor are not obvious.

## BLC

The BLC module provides the sensor-related BLC function, and configures the offsets of the R, Gr, Gb, and B component.

## DPC

The DPC module corrects static and dynamic defect pixels.

This module automatically detects and corrects dynamic defect pixels in realtime based on the configured threshold, and supports defect pixel cluster correction within 2 x 2 pixels in a single channel.

## GE

The GE module equalizes the Gb and Gr channels when imbalance occurs and improves the image quality in some scenarios.

## WDR

The WDR module provides the 2F-WDR frame mode. Details in bright and dark regions can still be observed in WDR scenarios.

## Expander

The expander module decompresses the compressed data in the sensor.

## Bayer NR

The Bayer NR (BNR) module implements image denoising in the Bayer domain. It aims to retain details while implementing denoising. This module can eliminate noises of the sensor based on the noise model provided by the user.

## LSC

The LSC module implements lens shading correction. Because of the optical characteristics of the lens, the luminous intensity of the edge area is lower than that of the central area in a sensor image. Therefore, gain compensation based on pixel position is required.



This module provides the gains of the R, Gr, Gb, and B components, and the precision of each gain is adjustable (controlled by the meshscale) by symmetrically dividing one frame into  $18 \times 32$  zones, providing gains of the window vertex, and obtaining other values through interpolation. The gain obtained after interpolation is multiplied by the corresponding pixel value, and the compensated image data is output.

## DG

The DG module provides the digital gain, and configures the offsets of the R, Gr, Gb, and B components (U8.8 precision).

## AE

The AE module collects AE statistics. Software adjusts the sensor based on statistics to implement AE. The AE statistics includes the zone value statistics of average R/Gr/Gb/B, global weighted average R/Gr/Gb/B, and 1024-segment histogram statistics.

- AE zonal statistics

The AE module supports a maximum of  $32 \times 32$  zones and a minimum of  $1 \times 1$  zone. Each zone can output the average R/Gr/Gb/B statistics. [Table 10-5](#) describes the mapping between the address and read data of zone statistics.  $m$  is a number of horizontal zones and  $n$  is a number of vertical zones.

**Table 10-5** AE zonal statistics

Memory Address	Zone	Memory Read Data			
		MEM_AVE_R_R_GR [31:16]	MEM_AVER_R_GR [15:0]	MEM_AVER_G_B [31:16]	MEM_AVER_G_B [15:0]
0	0	Average R	Average Gr	Average Gb	Average B
1	1	Average R	Average Gr	Average Gb	Average B
2	2	Average R	Average Gr	Average Gb	Average B
3	3	Average R	Average Gr	Average Gb	Average B
...					
$m^*n-3$	$m^*n-3$	Average R	Average Gr	Average Gb	Average B
$m^*n-2$	$m^*n-2$	Average R	Average Gr	Average Gb	Average B
$m^*n-1$	$m^*n-1$	Average R	Average Gr	Average Gb	Average B

- Weighted AE global statistics



The AE global statistics are basically the same as the AE zoned statistics. The AE module provides four types of global statistics. The physical meanings of the four types of global statistics are the same as those of the zoned statistics

- Weighted AE histogram statistics

The AE module provides the 1024-segment histogram statistics. [Table 10-6](#) describes the mapping between the address and read data of the histogram statistics for zone 0–1023.

**Table 10-6 AE AE histogram statistics**

Memory Address	Zone	Memory Read Data	
		[31:29]	[28:0]
0	0	0	Number of pixels whose value is 0
1	1	0	Number of pixels whose value is 1
2	2	0	Number of pixels whose value is 2
3	3	0	Number of pixels whose value is 3
...			
1022	1022	0	Number of pixels whose value is 1022
1023	1023	0	Number of pixels whose value is 1023

## AF Statistics

The AF module collects statistics on definition evaluation information. The number of image zones is configurable. This module supports a maximum of 17 x 15 zones. The minimum zone size is 32 x 32 and the maximum zone size is 512 x 512. The definition evaluation information about each zone is provided.

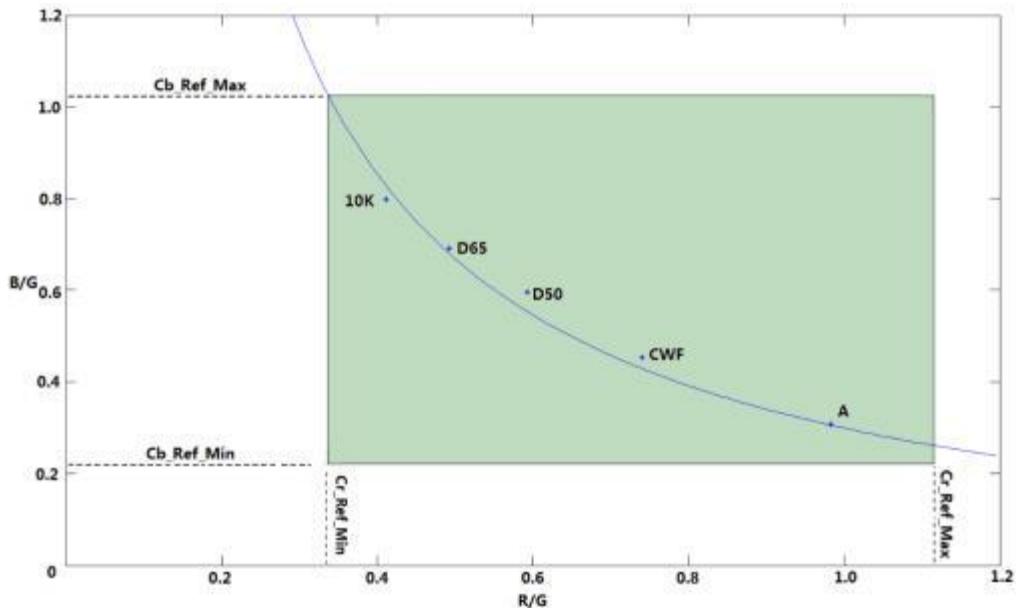
## AWB

The AWB statistics includes the global statistics and zonal statistics.

- Global statistics: average values of the R, G, and B components (average R/G/B) for an entire image and number of valid statistical pixels
- Zonal statistics are collected as follows:

- The maximum number of zones is 32 x 32. Each zone can output average RGB value and the number of valid statistical pixels.
- The valid statistical pixel is defined as that the luminance of the pixel meets the requirements of the upper limit and lower limit of the configured RGB as well as the configured color space limit.

**Figure 10-6** Color space of valid AWB pixel



- AWB zonal statistics

The AWB module supports a maximum of 32 x 32 zones. Each zone can output the average R/G/B. The width of the pixels involved in statistics is 16 bits. There are three types of statistics on the statistical pixels of a zone: number of valid statistical pixels (Count All). The preceding statistics are normalized to 16 bits based on the image size. [Table 10-7](#) describes the mapping between the address and read data of zone statistics.

**Table 10-7** AWB zonal statistics for mode A

<b>Memory Address</b>	<b>Zone</b>	<b>Memory Read Data</b>	
		<b>[31:16]</b>	<b>[15:0]</b>
0	0	Average G	Average R
1	0	Count All	Average B
2	1	Average G	Average R
...			



Memory Address	Zone	Memory Read Data	
		[31:16]	[15:0]
2046	1023	Average G	Average R
2047	1023	Count All	Average B

- AWB global statistics

The AWB global statistics are similar to the AWB zonal statistics. The AWB module provides global statistics of four registers. The physical meanings are the same as those of the zonal statistics.

## WB

The WB module provides the white balance function, and configures the gains of the R, Gr, Gb, and B components (U4.8 precision).

## DRC

The DRC module adjusts the display dynamic range of the image so that the image display effect on the display device is consistent with the image observed by human eyes.

## CAC

The CAC module is used to correct the axial chromatic aberration (purple fringing) and lateral chromatic aberration (color fringing on opposite sides of an object with different colors) introduced by the lens.

## DEMOSAIC

The demosaic module converts the raw image in Bayer format into the RGB image.

This module analyzes the characteristics of the internal image edge, uses pixel relevance interpolation, and effectively suppresses the anti-false color while ensuring high resolution and definition.

## CCM

The CCM module implements linear correction of the color space by using the standard 3 x 3 matrix and vector offset.

The firmware dynamically calculates the CCM coefficients of the current image based on the several pre-calculated groups of CCM coefficients and image color temperature. The firmware can also use the CCM to dynamically adjust the saturation based on the luminance of the current image.

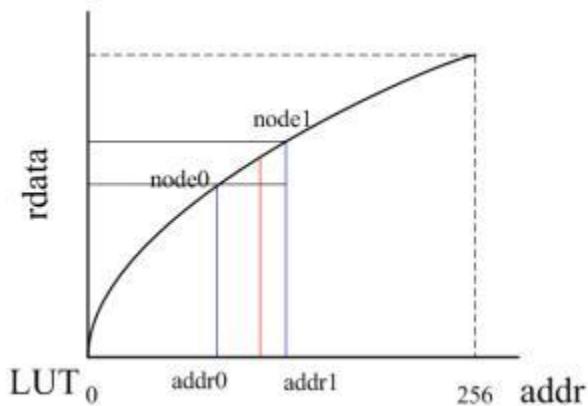
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} \text{coef } 00 & \text{coef } 01 & \text{coef } 02 \\ \text{coef } 10 & \text{coef } 11 & \text{coef } 12 \\ \text{coef } 20 & \text{coef } 21 & \text{coef } 22 \end{pmatrix} \begin{pmatrix} \text{根} \\ G + \text{in\_dc1} \\ B + \text{in\_dc2} \end{pmatrix} + \begin{pmatrix} \text{out\_dc0} \\ \text{out\_dc1} \\ \text{out\_dc2} \end{pmatrix}$$

In the preceding formula, R, G, and B indicate the input data, and coefis a  $3 \times 3$  configurable matrix coefficient. The coefficients are 13-bit numbers in S5.8 format.

## GAMMA

The gamma/preGamma module applies to RGB or Bayer domain and outputs the gamma/preGamma adjustment result. This module adjusts the luminance based on the gamma curve. The color channels are adjusted according to the gamma curve.

**Figure 10-7** Gamma curve



The gamma curve consists of 257 nodes (node 0 to node 256). The curve value corresponding to each node is a 12-bit unsigned number. The pixels between two nodes can be obtained through interpolation. The curve value of node 0 (gamma[0]) is 0 and the curve value of node 256 (gamma[256]) is 0xFFFF.

## DEHAZE

The dehaze module provides the powerful zoned anti-fog function and improves video contrast and definition in the haze scenario.

This module analyzes the image characteristics of each region, obtains the contrast specifications of each region, and implements pixel enhancement in each region. The enhancement strength of a pixel depends on the contrast specifications of the region this pixel belongs to and those of the adjacent regions.

## CSC

The CSC module converts the {R, G, B} input into {Y, U, V} by using the standard 3 x 3 matrix and vector offset.

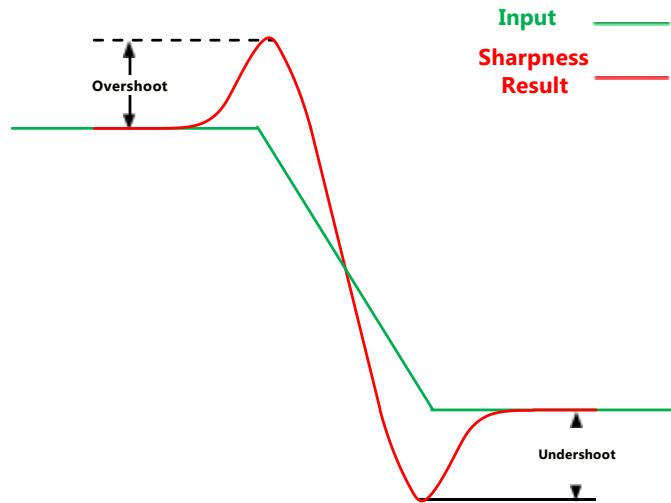
$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} \text{coef00} & \text{coef01} & \text{coef02} \\ \text{coef10} & \text{coef11} & \text{coef12} \\ \text{coef20} & \text{coef21} & \text{coef22} \end{pmatrix} \begin{pmatrix} R + \text{in\_dc0} \\ G + \text{in\_dc1} \\ B + \text{in\_dc2} \end{pmatrix} + \begin{pmatrix} \text{out\_dc0} \\ \text{out\_dc1} \\ \text{out\_dc2} \end{pmatrix}$$

The parameters can be changed based on format conversion requirements.

## SHARPEN

The sharpen module implements image sharpening to improve image definition. The sharpening strength can be adjusted by configuring the control parameters **edge\_amt** and **sharp\_amt**. Note that noises may be amplified if the sharpening strength is too high. Software adjusts parameter configurations based on the ISO value to achieve a balance between the definition and noise suppression while improving the visual effect of the image.

**Figure 10-8** Overshoot and undershoot occurrence during sharpening



In addition, white borders and black borders may appear due to the large positive and negative edge amplification caused by sharpening, as shown in [Figure 10-8](#). The white borders and black borders can be suppressed by adjusting the overshoot and undershoot control parameters.

## CA

The CA module provides the saturation adjustment function.



## LDCI

Based on local histogram equalization, the LDCI can enhance local saturation by improving detail effect in dark regions and enhancing the high frequency in the image.

## CDS

The CDS module converts the YUV444 into YUV422 or YUV420, and implements multi-order horizontal chrominance filtering. When the YUV420 output is required, this module implements average down sampling in the vertical direction of the chrominance. The visual loss caused by image conversion can be minimized by setting the filtering parameters to appropriate values.



# Contents

<b>11 Audio Interfaces .....</b>	<b>11-1</b>
<b>11.1 AIAO.....</b>	<b>11-1</b>
<b>11.1.1 Overview .....</b>	<b>11-1</b>
<b>11.1.2 Features.....</b>	<b>11-2</b>
<b>11.1.3 Function Description.....</b>	<b>11-3</b>
<b>11.1.4 Operating Mode.....</b>	<b>11-6</b>
<b>11.1.5 Register Summary .....</b>	<b>11-9</b>
<b>11.1.6 Register Description.....</b>	<b>11-11</b>
<b>11.2 Audio Codec .....</b>	<b>11-45</b>
<b>11.2.1 Overview .....</b>	<b>11-45</b>
<b>11.2.2 Features.....</b>	<b>11-45</b>
<b>11.2.3 Register Summary .....</b>	<b>11-46</b>
<b>11.2.4 Register Description.....</b>	<b>11-46</b>



## Figures

<b>Figure 11-1</b> Block diagram of the AIAO.....	11-1
<b>Figure 11-2</b> AIP 0 and AOP 0/AOP 1 connecting to an external ADC/DAC in I <sub>2</sub> S/PCM master/slave mode.....	11-3
<b>Figure 11-3</b> I <sub>2</sub> S interface timing .....	11-4
<b>Figure 11-4</b> Timing of the PCM interface in standard mode (PCM_OFFSET =1) .....	11-4
<b>Figure 11-5</b> Timing of the PCM interface in customized mode (PCM_OFFSET =0) .....	11-4
<b>Figure 11-6</b> I <sub>2</sub> S receiving 1-/2-/4-channel data .....	11-5
<b>Figure 11-7</b> 1-/2-/4-channel data reception in PCM mode when the frame rate is 256 fps and the offset value is 1 .....	11-6



## Tables

<b>Table 11-1</b> Variable in the register offset addresses .....	11-9
<b>Table 11-2</b> Summary of AIAO registers (base address: 0x17C0_0000) .....	11-9
<b>Table 11-3</b> Summary of AudioCodec registers (base address: 0x17C4_0000) .....	11-46

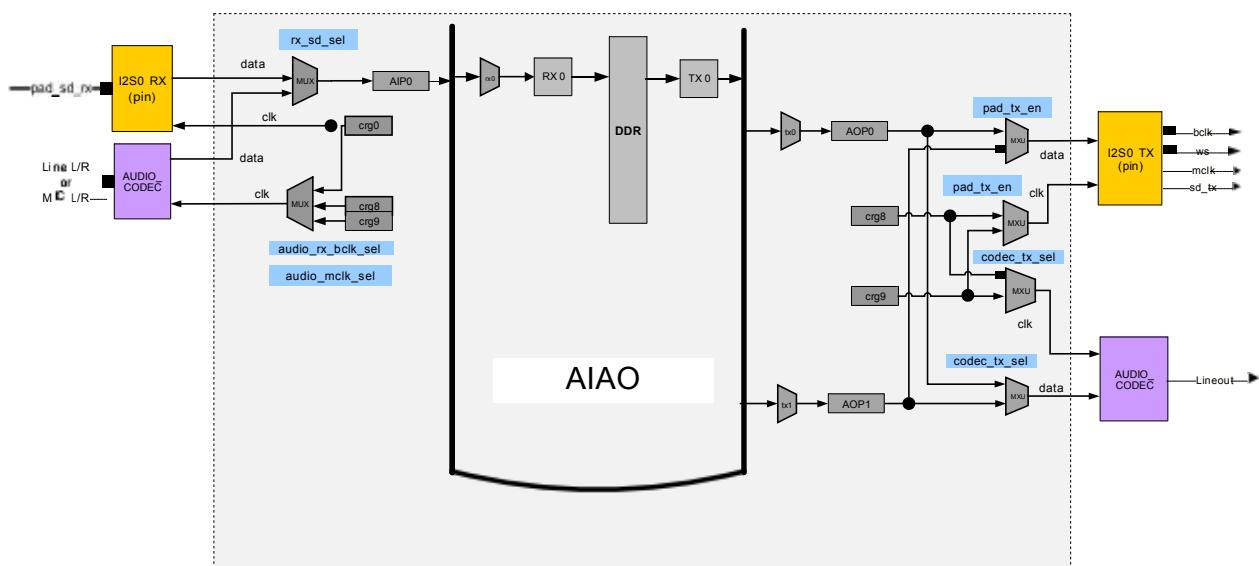
# 11 Audio Interfaces

## 11.1 AIAO

### 11.1.1 Overview

The audio input/output (AIAO) interface connects to the on-chip/off-chip audio coder/decoder (codec) to input and output audio data, implementing the recording, talkback, and playback functions. The chip provides one built-in AIAO interface that includes one audio input port (AIP) and two audio output ports (AOPs), supporting stereo inputs and outputs. AOP 0 and AOP 1 are connected to the I<sup>2</sup>S pins, built-in codec, and HDMI in the chip by using the MUX selection output. [Figure 11-1](#) shows the block diagram of the basic modules.

**Figure 11-1** Block diagram of the AIAO





### NOTE

- Texts highlighted in yellow indicate the channels related to the external codec.
- Texts highlighted in purple indicate the channels related to the internal codec.
- Texts highlighted in gray indicate internal logic.
- Texts highlighted in blue indicate the [AIAO\\_MUX\\_SEL](#) registers.
- To transmit both channels of AOP data, the data must be output from the same source.

## 11.1.2 Features

The AIAO interface can connect to the external codec in I<sup>2</sup>S and pulse code modulation (PCM) modes.

- The AIAO interface supports the I<sup>2</sup>S mode and interconnection to the on-chip audio codec. For details about the register configurations of the on-chip audio codec, see section [11.2 Audio Codec](#).
- Input (AIP 0) and output (AOP 0/AOP 1) are operated in the DMA mode. The storage and reading of data is realized by the cyclic buffer created by the software. The size and waterline of the cyclic buffers are configurable.

### PCM Mode

The PCM mode has the following features:

- Supports the master and slave mode.
- Transmits and receives 8-/16-bit PCM data in mono mode in normal mode.
- Receives 2-/4-channel 8-/16-bit PCM data in TDM mode.
- Supports the sampling rate ranging from 8 kHz to 48 kHz.
- Supports only short pulse sync signals in frame sync signals (the duration of those sync signals is one clock cycle). The PCM mode can work in both the standard mode and customized mode.
- Separately enables or disables input (AIP 0) and output (AOP 0/AOP 1).
- Uses the DMA operation for both RX (AIP 0) and TX (AOP 0/AOP 1). Data is accessed through the cyclic buffer created by software. The size and threshold of the cyclic buffer are adjustable.

### I<sup>2</sup>S Mode

The I<sup>2</sup>S mode has the following features:

- Supports the master and slave mode.
- AIP 0 can receive 1-/2-channel 16-/24-bit data in normal mode.
- AIP 0 can receive 2-/4-channel 8-/16-bit data in TDM mode.
- AOP 0 and AOP 1 can send 1-/2-channel 16-/24-bit data.
- Supports the sampling rate ranging from 8 kHz to 48 kHz.



- Separately enables or disables input (AIP 0) and output (AOP 0/AOP 1).
- Uses the DMA operation for both RX (AIP 0) and TX (AOP 0/AOP 1). Data is accessed through the cyclic buffer created by software. The size and threshold of the cyclic buffer are adjustable.

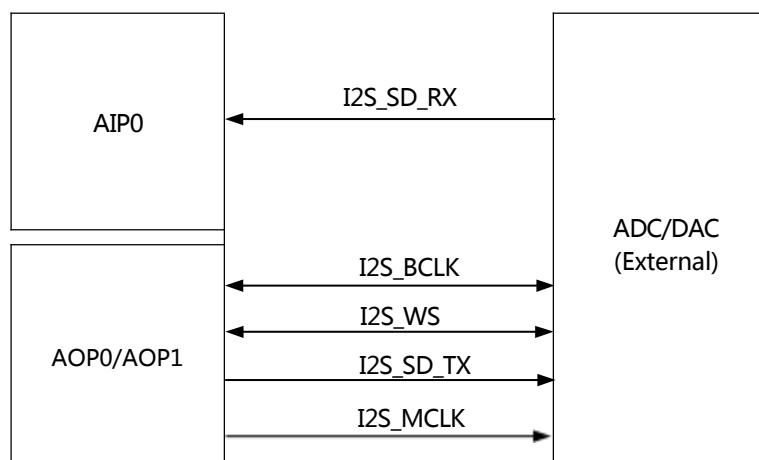
### 11.1.3 Function Description

#### Typical Application

The chip has one built-in AIP and two AOPs.

- When AIP 0, AOP 0, or AOP 1 is connected to the internal audio codec, only the I<sup>2</sup>S master mode is supported.
- AIP 0 can connect to an external audio codec in I<sup>2</sup>S/PCM master/slave mode.
- AOP 0/AOP 1 can connect to an external audio codec in I<sup>2</sup>S/PCM master/slave mode.

**Figure 11-2** AIP 0 and AOP 0/AOP 1 connecting to an external ADC/DAC in I<sup>2</sup>S/PCM master/slave mode



#### Function Principle

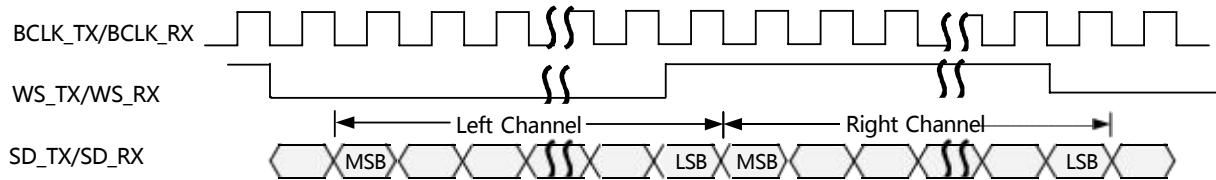
AIP 0 receives the audio data converted by the external ADC or built-in audio codec through analog-to-digital (AD) conversion over the I<sup>2</sup>S interface, and saves the data to the cyclic buffer of AIP 0. Then, the CPU obtains and stores the data. In this way, the recording function is implemented.

AOP 0/1 reads audio data from the cyclic buffer. Then, AOP 0/1 transmits the audio data to the external DAC or built-in audio codec through the I<sup>2</sup>S or PCM interface at a specified sampling rate for DA conversion.

[Figure 11-3](#) shows the I<sup>2</sup>S timing supported when an external I<sup>2</sup>S interface is connected. According to the protocol, the falling edge of the BCLK\_TX/BCLK\_RX

clock is used for TX, and the rising edge is used for data sampling. The most significant bit (MSB) is valid in the next clock cycle when WS\_TX/WS\_RX changes. The data is always transferred in the sequence of from the MSB to the least significant bit (LSB).

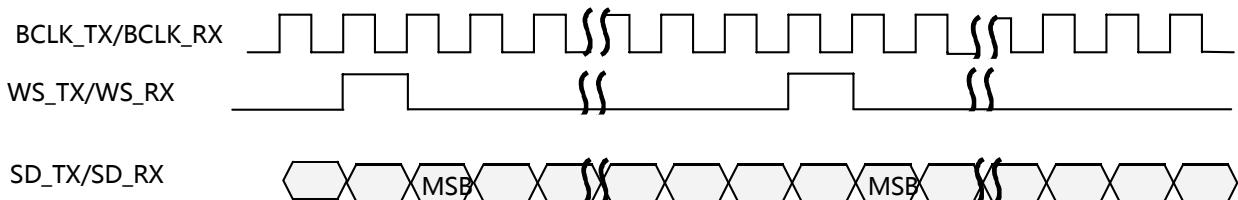
**Figure 11-3** I<sup>S</sup> interface timing



The PCM interface transmits mono data, and WS\_TX/WS\_RX identifies the start position of the data. The MSB is transmitted (received) first. The falling edge of the clock is used for TX, and the rising edge is used for data sampling (clock inverse allowed). In standard timing mode, the MSB is valid within the next cycle after the high-level pulse of WS\_TX/WS\_RX. In customized timing mode, the position of the MSB can be adjusted based on PCM\_OFFSET. When PCM\_OFFSET is 0, the MSB is aligned with the high-level pulse of WS\_TX/WS\_RX.

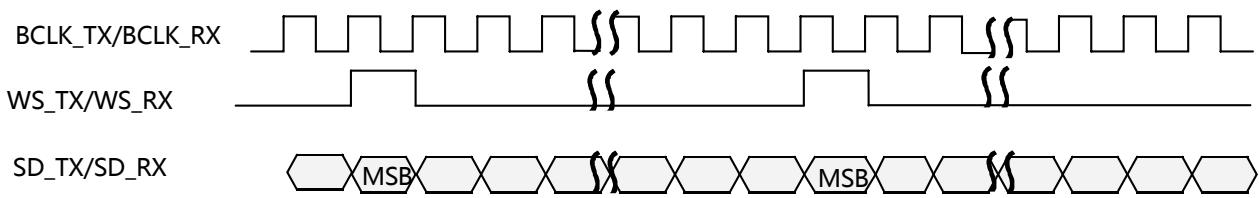
[Figure 11-4](#) shows the timing diagram of the PCM interface in standard mode.

**Figure 11-4** Timing of the PCM interface in standard mode (PCM\_OFFSET =1)



[Figure 11-5](#) shows the timing of the PCM interface in customized mode.

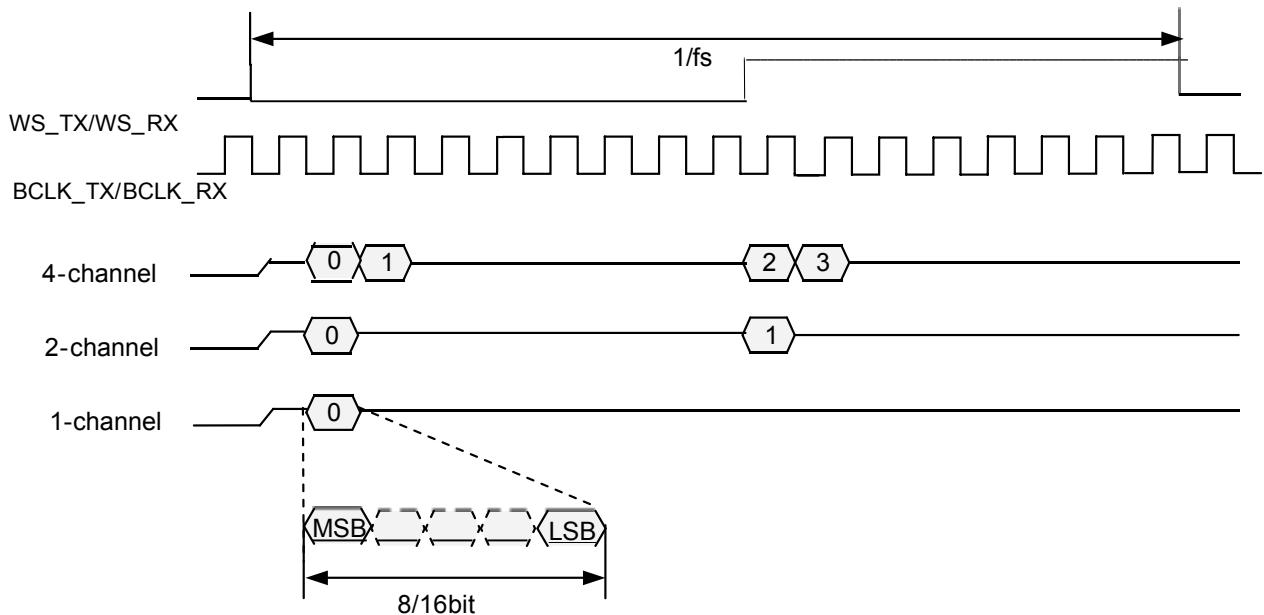
**Figure 11-5** Timing of the PCM interface in customized mode (PCM\_OFFSET =0)



In customized mode, data and WS pulses start to be sent when they are in the same cycle.

When receiving the 2-/4-channel 8-bit or 16-bit data, the I<sup>2</sup>S interface stores the data into the left and right audio channels of the I<sup>2</sup>S timing, respectively, as shown in [Figure 11-6](#).

**Figure 11-6** I<sup>2</sup>S receiving 1-/2-/4-channel data

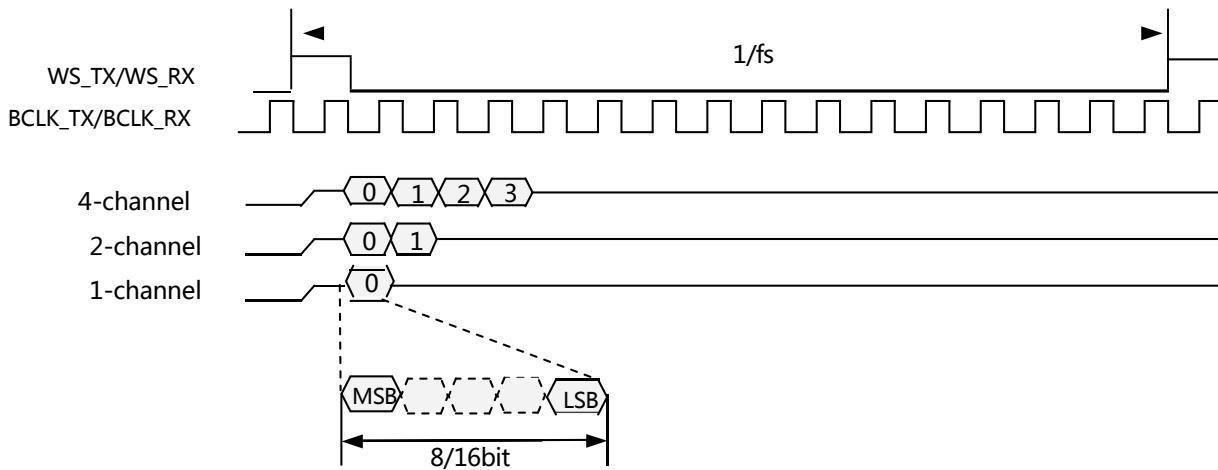


#### NOTE

The 1-channel data input is supported only in non-TDM mode.

[Figure 11-7](#) shows the multi-channel data reception in PCM mode. Both the standard and customized PCM modes are supported. The AIAO can choose to receive data at the rising edge or falling edge.

**Figure 11-7** 1-/2-/4-channel data reception in PCM mode when the frame rate is 256 fps and the offset value is 1



#### NOTE

The 1-channel data input is supported only in non-TDM mode.

### 11.1.4 Operating Mode

#### Clock Gating and Configuration

Before enabling the AIAO interface for audio recording or playback, you must enable clock gating of the related channels (AIP/AOP). To enable clock gating, perform the following steps:

- Step 1** In the PERI\_CRG10784 system register, set **aiao\_pll\_cksel** to **0** and select the 2376 MHz clock source.
- Step 2** In the PERI\_CRG10784 system register, set **aiao\_srst\_req** to **0** to deassert the AIAO reset, and then set **aiao\_pll\_cken** and **aiao\_cken** to **1** to enable the clock gating.
- Step 3** In the **AIAO\_MUX\_SEL** register of the AIAO, configure the MUX relationship between the clock and data.
- Step 4** Configure frequency dividers.

Configure the **I2S\_CRG\_CFG0\_00/I2S\_CRG\_CFG1\_00/I2S\_CRG\_CFG0\_08**, **I2S\_CRG\_CFG1\_08/I2S\_CRG\_CFG0\_09**, and **I2S\_CRG\_CFG1\_09** AIAO registers, and select proper frequency dividers. For the mapping between the CRG and the AIP/AOP, see [Figure 11-7](#).

----End



## Soft Reset

The three internal channels (AIP 0, AOP 0, and AOP 1) of the AIAO module support separate soft reset. When the AIAO module is reset, the three channels are reset at the same time.

## Recording Process

The following example assumes that the audio channels are stereo channels in I<sup>2</sup>S mode, the sampling rate is 48 kHz, the sampling precision is 16 bits, the AIAO PLL clock source is a 2376 MHz clock, and the AIAO clock of the system controller is enabled. To record data, perform the following steps:

- Step 1** Set `I2S_CRG_CFG0_08` to 0x000A9778 to set the channel-8 MCLK to 12.288 MHz.
- Step 2** Set `I2S_CRG_CFG1_08` to 0x00000133 to enable the channel-8 clock, set the BCLK to the divide-by-4 clock of the MCLK, and set the FCLK to the divide-by-64 clock of the BCLK. The FCLK is 48 kHz.
- Step 3** Set `AIAO_SWITCH_RX_BCLK` to 0x00000008 to set the working clock of the RX channel to channel-8 clock.
- Step 4** Set `RX_IF_ATTR` to 0xE4800014 to set the operating mode of the RX channel to I<sup>2</sup>S stereo mode and sampling precision to 16 bits.
- Step 5** Set `RX_BUFF_SADDR_LOW` to configure of the start address for the DDR allocation (for example, 0x80000000). Set `RX_BUFF_SIZE` to the size of the allocated DDR buffer (for example, 0x0000f000). Set `RX_BUFF_WPTR` and `RX_BUFF_RPTR` to 0x0 to initialize the read and write pointers. Set `RX_TRANS_SIZE` (for example, 0x00000f00).
- Step 6** Configure `RX_INT_ENA` to enable the corresponding interrupt of the RX channel as required. For example, set it to 0x00000001 to enable the trans\_int interrupt.
- Step 7** Set `RX_DSP_CTRL` to 0x10000000 to enable the RX channel. Then the RX channel starts recording.
- Step 8** Read `RX_BUFF_WPTR` and `RX_BUFF_RPTR` to check the empty/full status of the cyclic buffer and the valid data amount. Ensure that data is fetched before the cyclic buffer is full and the updated read address of the cyclic buffer is written to `RX_BUFF_RPTR`. Otherwise, an overflow occurs in the cyclic buffer and the audio is discontinuous.
- Step 9** After recording, write 0x00000000 to `RX_DSP_CTRL` and query `RX_DSP_CTRL` until its value is 0x20000000, which indicates that the RX channel stops receiving data.

----End



### NOTICE

Configure AIP 0 clock before starting AIP 0, ensuring that AIAO\_BCLK\_RX and AIAO\_WS\_RX are normal.

## Playback Process

The playback processes for the AOP 0 and AOP 1 are the same. The following uses AOP 0 as an example and assumes that the audio channels are stereo channels in I<sup>2</sup>S mode, the sampling rate is 48 kHz, the sampling precision is 16 bits, the AIAO PLL clock source is a 2376 MHz clock, and the AIAO clock of the system controller is enabled. To play data, perform the following steps:

- Step 1** Set [I2S\\_CRG\\_CFG0\\_08](#) to 0x000A9778 to set the channel-8 MCLK to 12.288 MHz.
- Step 2** Set [I2S\\_CRG\\_CFG1\\_08](#) to 0x00000133 to enable the channel-8 clock, set the BCLK to the divide-by-4 clock of the MCLK, and set the FCLK to the divide-by-64 clock of the BCLK. The FCLK is 48 kHz.
- Step 3** Set [AIAO\\_SWITCH\\_TX\\_BCLK](#) to 0x00000008 to set the working clock of the TX channel to channel-8 clock.
- Step 4** Set [TX\\_IF\\_ATTR](#) to 0xE4000014 to set the operating mode of the TX channel to I<sup>2</sup>S stereo mode and sampling precision to 16 bits.
- Step 5** Set [TX\\_BUFF\\_SADDR\\_LOW](#) to set the start address for the buffer allocation (for example, 0x80000000). Set [TX\\_BUFF\\_SIZE](#) to the size of the allocated buffer. Set [TX\\_BUFF\\_WPTR](#) and [TX\\_BUFF\\_RPTR](#) to 0x0 to initialize the read and write pointers. Set [TX\\_TRANS\\_SIZE](#). For details, see step 5 in the "Recording Process" section.
- Step 6** Configure [TX\\_INT\\_ENA](#) to enable the corresponding interrupt of the RX channel as required. For example, set it to 0x00000001 to enable the trans\_int interrupt.
- Step 7** Set [TX\\_DSP\\_CTRL](#) to 0x10000000 to enable the playback channel.
- Step 8** Read [TX\\_BUFF\\_WPTR](#) and [TX\\_BUFF\\_RPTR](#) to check the empty/full status of the cyclic buffer and the valid data amount. Ensure that new audio data is stuffed before the cyclic buffer is empty and the updated write address for the cyclic buffer is written to [TX\\_BUFF\\_WPTR](#). Otherwise, an underflow occurs in the cyclic buffer and the audio is discontinuous.
- Step 9** After playback, set [TX\\_DSP\\_CTRL](#) to 0x00000000 to stop the playback channel and query [TX\\_DSP\\_CTRL](#) until its value is 0x20000000, which indicates that the playback channel stops working.

----End



## NOTICE

- Configure AOP 0 clock before starting AOP 0, ensuring that AIAO\_BCLK\_TX and AIAO\_WS\_TX are normal. This requirement also applies to AOP 1.
- Ensure that the available space of the AOP 0 cyclic buffer is greater than or equal to 32 bytes when writing data to the cyclic buffer and updating [TX\\_BUFF\\_WPTR](#). This requirement also applies to AOP 1.

### 11.1.5 Register Summary

[Table 11-1](#) describes the value range and meaning of the variable in the offset addresses of registers.

**Table 11-1** Variable in the register offset addresses

Variable	Value Range	Description
m	0	TX channel ID
n	0	RX channel ID

[Table 11-2](#) describes AIAO registers.

**Table 11-2** Summary of AIAO registers (base address: 0x17C0\_0000)

Offset Address	Register	Description	Page
0x0000	AIAO_INT_ENA	AIAO interrupt enable register	<a href="#">11-12</a>
0x0004	AIAO_INT_STATUS	AIAO interrupt status register	<a href="#">11-12</a>
0x0008	AIAO_INT_RAW	AIAO raw interrupt register	<a href="#">11-12</a>
0x0028	AIAO_SWITCH_RX_B_CLK	AIAO I <sup>S</sup> RX BCLK switch configuration register	<a href="#">11-13</a>
0x002C	AIAO_SWITCH_TX_B_CLK	AIAO I <sup>S</sup> TX BCLK switch configuration register	<a href="#">11-13</a>
0x0030	AIAO_STATUS	AIAO status register	<a href="#">11-14</a>
0x006C	AIAO_MUX_SEL	I <sup>S</sup> channel select control register	<a href="#">11-14</a>
0x0100	I2S_CRG_CFG0_00	I <sup>S</sup> 00 CRG configuration register 0	<a href="#">11-16</a>
0x0104	I2S_CRG_CFG1_00	I <sup>S</sup> 00 CRG configuration register 1	<a href="#">11-16</a>
0x0140	I2S_CRG_CFG0_08	I <sup>S</sup> 08 CRG configuration register 0	<a href="#">11-18</a>
0x0144	I2S_CRG_CFG1_08	I <sup>S</sup> 08 CRG configuration register 1	<a href="#">11-19</a>



Offset Address	Register	Description	Page
0x0148	I2S_CRG_CFG0_09	PS 09 CRG configuration register 0	<a href="#">11-21</a>
0x014C	I2S_CRG_CFG1_09	PS 09 CRG configuration register 1	<a href="#">11-21</a>
0x1000+0x 100×n	RX_IF_ATTRI	Interface attribute configuration register for the RX channel	<a href="#">11-24</a>
0x1004+0x 100×n	RX_DSP_CTRL	RX channel control register	<a href="#">11-26</a>
0x1080+0x 100×n	RX_BUFF_SADDR_LO W	DDR buffer start address register for the RX channel	<a href="#">11-26</a>
0x1084+0x 100×n	RX_BUFF_SIZE	DDR buffer size register for the RX channel	<a href="#">11-26</a>
0x1088+0x 100×n	RX_BUFF_WPTR	DDR buffer write address register for the RX channel	<a href="#">11-27</a>
0x108C+0x 100×n	RX_BUFF_RPTR	DDR buffer read address register for the RX channel	<a href="#">11-27</a>
0x1090+0x 100×n	RX_BUFF_ALFULL_T H	DDR buffer almost full threshold register for the RX channel	<a href="#">11-28</a>
0x1094+0x 100×n	RX_TRANS_SIZE	Data transfer length register for the RX channel	<a href="#">11-28</a>
0x1098+0x 100×n	RX_WPTR_TMP	Write address storage register for the RX channel upon reporting of the transfer completion interrupt	<a href="#">11-29</a>
0x10A0+0x 100×n	RX_INT_ENA	Interrupt enable register for the RX channel	<a href="#">11-29</a>
0x10A4+0x 100×n	RX_INT_RAW	Raw interrupt register for the RX channel	<a href="#">11-30</a>
0x10A8+0x 100×n	RX_INT_STATUS	Interrupt status register for the RX channel	<a href="#">11-31</a>
0x10AC+0x 100×n	RX_INT_CLR	Interrupt clear register for the RX channel	<a href="#">11-32</a>
0x2000+0x 100×m	TX_IF_ATTRI	Interface attribute configuration register for the TX channel	<a href="#">11-33</a>
0x2004+0x 100×m	TX_DSP_CTRL	TX channel control register	<a href="#">11-35</a>



Offset Address	Register	Description	Page
0x2020+0x 100×m	TX_WS_CNT	WS cyclic count status register for the TX channel	<a href="#">11-37</a>
0x2024+0x 100×n	TX_BCLK_CNT	BCLK cyclic count status register for the TX channel	<a href="#">11-37</a>
0x2080+0x 100×m	TX_BUFF_SADDR_LO_W	DDR buffer start address register for the TX channel	<a href="#">11-38</a>
0x2084+0x 100×m	TX_BUFF_SIZE	DDR buffer size register for the TX channel	<a href="#">11-38</a>
0x2088+0x 100×m	TX_BUFF_WPTR	DDR buffer write address register for the TX channel	<a href="#">11-38</a>
0x208C+0x 100×m	TX_BUFF_RPTR	DDR buffer read address register for the TX channel	<a href="#">11-39</a>
0x2090+0x 100×m	TX_BUFF_ALEMPY_TH	DDR buffer almost empty threshold register for the TX channel	<a href="#">11-39</a>
0x2094+0x 100×m	TX_TRANS_SIZE	Data transfer length register for the TX channel	<a href="#">11-40</a>
0x2098+0x 100×m	TX_RPTR_TMP	Read address storage register for the TX channel upon reporting of the transfer completion interrupt	<a href="#">11-40</a>
0x20A0+0x 100×m	TX_INT_ENA	Interrupt enable register for the TX channel	<a href="#">11-40</a>
0x20A4+0x 100×m	TX_INT_RAW	Raw interrupt register for the TX channel	<a href="#">11-41</a>
0x20A8+0x 100×m	TX_INT_STATUS	Interrupt status register for the TX channel	<a href="#">11-43</a>
0x20AC+0x 100×m	TX_INT_CLR	Interrupt clear register for the TX channel	<a href="#">11-44</a>

## 11.1.6 Register Description

### NOTE

In the register description, BCLK refers to a bit clock, MCLK refers to the master clock, and FSCLK refers to a sampling clock.



## AIAO\_INT\_ENA

AIAO\_INT\_ENA is the AIAO interrupt enable register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	Reserved	0x0000
[16]	RW	tx_ch0_int_ena	TX channel 0 interrupt enable 0: disabled 1: enabled	0x0
[15:1]	-	reserved	Reserved	0x0000
[0]	RW	rx_ch0_int_ena	RX channel 0 interrupt enable 0: disabled 1: enabled	0x0

## AIAO\_INT\_STATUS

AIAO\_INT\_STATUS is the AIAO interrupt status register.

Offset Address: 0x0004 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:17]	-	reserved	Reserved	0x0000
[16]	RO	tx_ch0_int_status	Interrupt status of TX channel 0 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[15:1]	-	reserved	Reserved	0x0000
[0]	RO	rx_ch0_int_status	Interrupt status of RX channel 0 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## AIAO\_INT\_RAW

AIAO\_INT\_RAW is the AIAO raw interrupt register.

Offset Address: 0x0008 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:17]	-	reserved	Reserved	0x0000
[16]	RO	tx_ch0_int_raw	Raw interrupt of TX channel 0 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[15:1]	-	reserved	Reserved	0x0000
[0]	RO	rx_ch0_int_raw	Raw interrupt of RX channel 0 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0

## AIAO\_SWITCH\_RX\_BCLK

AIAO\_SWITCH\_RX\_BCLK is the AIAO I<sup>S</sup> RX BCLK switch configuration register.

Offset Address: 0x0028 Total Reset Value: 0x0000\_0070

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000007
[3:0]	RW	inner_bclk_ws_sel_rx_00	Internal bclk select for RX channel 0 0x0: bclk 0 0x8: bclk 8 0x9: bclk 9 Other values: reserved	0x0

## AIAO\_SWITCH\_TX\_BCLK

AIAO\_SWITCH\_TX\_BCLK is the AIAO I<sup>S</sup> TX BCLK switch configuration register.

Offset Address: 0x002C Total Reset Value: 0x0000\_00F8

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000F
[3:0]	RW	inner_bclk_ws_sel_tx_00	Internal BCLK select for TX channel 0 0x0: bclk0 0x8: bclk8 0x9: bclk9	0x8



Bits	Access	Name	Description	Reset
			Other values: reserved	

## AIAO\_STATUS

AIAO\_STATUS is an AIAO status register.

Offset Address: 0x0030 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	srst_rdy	APB reset deassertion indicator 0: resetting 1: reset completed	0x1

## AIAO\_MUX\_SEL

AIAO\_MUX\_SEL is the I<sup>2</sup>S channel selection control register.

Offset Address: 0x006C Total Reset Value: 0x0003\_9000

Bits	Access	Name	Description	Reset
[31:23]	-	reserved	Reserved	0x000
[22]	RW	cr9_mode_sel	FSCLK generation mode control register in crg9 master mode 0: I <sup>2</sup> S mode 1: PCM mode	0x0
[21:18]	RW	i2s1_tx_sd_src_sel	tx1_sd source select 0x0: I <sup>2</sup> S TX0 0x8: I <sup>2</sup> S RX0 0x9: I <sup>2</sup> S RX1 Other values: reserved	0x0
[17:16]	RW	i2s1_tx_sd3_sel	tx1_sd sd3 select 00: sd0 01: sd1 10: sd2 11: sd3	0x3



Bits	Access	Name	Description	Reset
			Note: Select the channel and data line in sequence.	
[15:14]	RW	i2s1_tx_sd2_sel	tx1_sd sd2 select 00: sd0 01: sd1 10: sd2 11: sd3 Note: Select the channel and data line in sequence.	0x2
[13:12]	RW	i2s1_tx_sd1_sel	tx1_sd sd1 select 00: sd0 01: sd1 10: sd2 11: sd3 Note: Select the channel and data line in sequence.	0x1
[11:10]	RW	i2s1_tx_sd0_sel	tx1_sd sd0 select 00: sd0 01: sd1 10: sd2 11: sd3 Note: Select the channel and data line in sequence.	0x0
[9:5]	-	reserved	Reserved	0x00
[4]	RW	codec_tx_sel	DAC input data and clock select 0: tx0_sd, crg8. 1: tx1_sd, crg9.	0x0
[3]	RW	audio_mclk_sel	MCLK select for the chip I <sup>S</sup> interface. 0: aiao_i2s_rx_mclk, that is, crg0 1: aiao_i2s_tx_mclk, that is, crg8 or crg9. The crg8 or crg9 clock is selected by configuring <b>codec_tx_sel</b> .	0x0
[2]	RW	rx_sd_sel	rx0_sd data source select 0: I <sup>S</sup> pin 1: iis_sd output of the built-in CODEC_ADC	0x0



Bits	Access	Name	Description	Reset
[1]	RW	audio_rx_bclk_sel	BCLK/WS select for the built-in CODEC_ADC 0: crg0 clock 1: The crg8 or crg9 clock is selected by configuring <b>codec_tx_sel</b> . The adc and dac share the same clock.	0x0
[0]	RW	pad_tx_en	Data and clock select for the iis_tx0 output pin 0: tx1_sd, crg9 1: tx0_sd, crg8	0x0

## I2S\_CRG\_CFG0\_00

I2S\_CRG\_CFG0\_00 is I<sup>2</sup>S 00 CRG configuration register 0.

Offset Address: 0x0100 Total Reset Value: 0x00AA\_AAAA

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26:0]	RW	aiao_mclk_div	Configured value of the frequency division clock of the MCLK. The configured value is calculated as follows: (MCLK/SIO clock source frequency) x 2 <sup>27</sup> .	0x0AAAAAA

## I2S\_CRG\_CFG1\_00

I2S\_CRG\_CFG1\_00 is I<sup>2</sup>S 00 CRG configuration register 1.

Offset Address: 0x0104 Total Reset Value: 0x0000\_0131

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15]	RW	aiao_ws_en	WS enable 0: disabled 1: enabled	0x0
[14]	RW	aiao_bclk_en	BCLK enable 0: disabled	0x0



Bits	Access	Name	Description	Reset
			1: enabled	
[13]	RW	aiao_bclkout_pctrl	BCLKOUT polarity control 0: forward 1: inverted	0x0
[12]	RW	aiao_bclkin_pctrl	BCLKIN polarity control 0: forward 1: inverted	0x0
[11]	RW	aiao_bclk_sel	BCLK/FCLK select 0: The clock is generated from inside. 1: The clock is input from outside.	0x0
[10]	RW	aiao_bclk_oen	BCLK/FSCLK I/O OEN control 0: The BCLK/FSCLK I/O is output. 1: The BCLK/FSCLK I/O is input. Note: This field must be used in conjunction with <b>aiao_bclk_sel</b> to select the master/slave mode of the I <sup>S</sup> interface.	0x0
[9]	RW	aiao_srst_req	Soft reset request 0: reset deasserted 1: reset	0x0
[8]	RW	aiao_cken	clock status 0: disabled 1: enabled	0x1
[7]	-	reserved	Reserved	0x0
[6:4]	RW	aiao_fsclk_div	Frequency division relationship between the BCLK and the sampling clock FSCLK 000: The FSCLK is obtained by dividing the BCLK by 16. 001: The FSCLK is obtained by dividing the BCLK by 32. 010: The FSCLK is obtained by dividing the BCLK by 48. 011: The FSCLK is obtained by dividing the BCLK by 64. 100: The FSCLK is obtained by dividing the BCLK by 128.	0x3



Bits	Access	Name	Description	Reset
			101: The FSCLK is obtained by dividing the BCLK by 256. Other values: The FSCLK is obtained by dividing the BCLK by 8.	
[3:0]	RW	aiao_bclk_div	Frequency division relationship between the MCLK and the BCLK 0x0: The BCLK is obtained by dividing the MCLK by 1. 0x1: The BCLK is obtained by dividing the MCLK by 3. 0x2: The BCLK is obtained by dividing the MCLK by 2. 0x3: The BCLK is obtained by dividing the MCLK by 4. 0x4: The BCLK is obtained by dividing the MCLK by 6. 0x5: The BCLK is obtained by dividing the MCLK by 8. 0x6: The BCLK is obtained by dividing the MCLK by 12. 0x7: The BCLK is obtained by dividing the MCLK by 16. 0x8: The BCLK is obtained by dividing the MCLK by 24. 0x9: The BCLK is obtained by dividing the MCLK by 32. 0xA: The BCLK is obtained by dividing the MCLK by 48. 0xB: The BCLK is obtained by dividing the MCLK by 64. Other values: The BCLK is obtained by dividing the MCLK by 8.	0x1

## I2S\_CRG\_CFG0\_08

I2S\_CRG\_CFG0\_08 is I<sup>2</sup>S 08 CRG configuration register 0.

Offset Address: 0x0140 Total Reset Value: 0x00AA\_AAAA



Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26:0]	RW	aiao_mclk_div	Configured value of the frequency division clock of the MCLK. The configured value is calculated as follows: (MCLK/SIO clock source frequency) x 2^27.	0x0AAAAAA

## I2S\_CRG\_CFG1\_08

I2S\_CRG\_CFG1\_08 is I<sup>2</sup>S 08 CRG configuration register 1.

Offset Address: 0x0144 Total Reset Value: 0x0000\_0131

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15]	RW	aiao_ws_en	WS enable 0: disabled 1: enabled	0x0
[14]	RW	aiao_bclk_en	BCLK enable 0: disabled 1: enabled	0x0
[13]	RW	aiao_bclkout_pctrl	BCLKOUT polarity control 0: forward 1: inverted	0x0
[12]	RW	aiao_bclkin_pctrl	BCLKin polarity control 0: forward 1: inverted	0x0
[11]	RW	aiao_bclk_sel	BCLK/FCLK select 0: The clock is generated from inside. 1: The clock is input from outside.	0x0
[10]	RW	aiao_bclk_oen	BCLK/FSCLK I/O OEN control 0: The BCLK/FSCLK I/O is output. 1: The BCLK/FSCLK I/O is input. Note: This field must be used in conjunction with <b>aiao_bclk_sel</b> to select the master/slave mode of the I <sup>2</sup> S	0x0



Bits	Access	Name	Description	Reset
			interface.	
[9]	RW	aiao_srst_req	Soft reset request 0: reset deasserted 1: reset	0x0
[8]	RW	aiao_cken	clock status 0: disabled 1: enabled	0x1
[7]	-	reserved	Reserved	0x0
[6:4]	RW	aiao_fsclk_div	Frequency division relationship between the BCLK and the sampling clock FSCLK 000: The FSCLK is obtained by dividing the BCLK by 16. 001: The FSCLK is obtained by dividing the BCLK by 32. 010: The FSCLK is obtained by dividing the BCLK by 48. 011: The FSCLK is obtained by dividing the BCLK by 64. 100: The FSCLK is obtained by dividing the BCLK by 128. 101: The FSCLK is obtained by dividing the BCLK by 256. Other values: The FSCLK is obtained by dividing the BCLK by 8.	0x3
[3:0]	RW	aiao_bclk_div	Frequency division relationship between the MCLK and the BCLK 0x0: The BCLK is obtained by dividing the MCLK by 1. 0x1: The BCLK is obtained by dividing the MCLK by 3. 0x2: The BCLK is obtained by dividing the MCLK by 2. 0x3: The BCLK is obtained by dividing the MCLK by 4. 0x4: The BCLK is obtained by dividing the MCLK by 6. 0x5: The BCLK is obtained by dividing the MCLK by 8.	0x1



Bits	Access	Name	Description	Reset
			0x6: The BCLK is obtained by dividing the MCLK by 12. 0x7: The BCLK is obtained by dividing the MCLK by 16. 0x8: The BCLK is obtained by dividing the MCLK by 24. 0x9: The BCLK is obtained by dividing the MCLK by 32. 0xA: The BCLK is obtained by dividing the MCLK by 48. 0xB: The BCLK is obtained by dividing the MCLK by 64. Other values: The BCLK is obtained by dividing the MCLK by 8.	

## I2S\_CRG\_CFG0\_09

I2S\_CRG\_CFG0\_09 is I<sup>S</sup> 09 CRG configuration register 0.

Offset Address: 0x0148 Total Reset Value: 0x00AA\_AAAA

Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x00
[26:0]	RW	aiao_mclk_div	Configured value of the frequency division clock of the MCLK. The configured value is calculated as follows: (MCLK/SIO clock source frequency) x 2 <sup>27</sup> .	0x0AAAAAAA

## I2S\_CRG\_CFG1\_09

I2S\_CRG\_CFG1\_09 is I<sup>S</sup> 09 CRG configuration register 1.

Offset Address: 0x014C Total Reset Value: 0x0000\_0131

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15]	RW	aiao_ws_en	WS enable 0: disabled	0x0



Bits	Access	Name	Description	Reset
			1: enabled	
[14]	RW	aiao_bclk_en	BCLK enable 0: disabled 1: enabled	0x0
[13]	RW	aiao_bclkout_pctrl	BCLKOUT polarity control 0: forward 1: inverted	0x0
[12]	RW	aiao_bclkin_pctrl	BCLKIN polarity control 0: forward 1: inverted	0x0
[11]	RW	aiao_bclk_sel	BCLK/FCLK select 0: The clock is generated from inside. 1: The clock is input from outside.	0x0
[10]	RW	aiao_bclk_oen	BCLK/FSCLK I/O oen control 0: The BCLK/FSCLK I/O is output. 1: The BCLK/FSCLK I/O is input. Note: This field must be used in conjunction with <b>aiao_bclk_sel</b> to select the master/slave mode of the I <sup>S</sup> interface.	0x0
[9]	RW	aiao_srst_req	Soft reset request 0: reset deasserted 1: reset	0x0
[8]	RW	aiao_cken	clock status 0: disabled 1: enabled	0x1
[7]	-	reserved	Reserved	0x0
[6:4]	RW	aiao_fsclk_div	Frequency division relationship between the BCLK and the sampling clock FSCLK 000: The FSCLK is obtained by dividing the BCLK by 16. 001: The FSCLK is obtained by dividing the BCLK by 32. 010: The FSCLK is obtained by dividing the BCLK by 48.	0x3



Bits	Access	Name	Description	Reset
			011: The FSCLK is obtained by dividing the BCLK by 64. 100: The FSCLK is obtained by dividing the BCLK by 128. 101: The FSCLK is obtained by dividing the BCLK by 256. Other values: The FSCLK is obtained by dividing the BCLK by 8.	
[3:0]	RW	aiao_bclk_div	Frequency division relationship between the MCLK and the BCLK 0x0: The BCLK is obtained by dividing the MCLK by 1. 0x1: The BCLK is obtained by dividing the MCLK by 3. 0x2: The BCLK is obtained by dividing the MCLK by 2. 0x3: The BCLK is obtained by dividing the MCLK by 4. 0x4: The BCLK is obtained by dividing the MCLK by 6. 0x5: The BCLK is obtained by dividing the MCLK by 8. 0x6: The BCLK is obtained by dividing the MCLK by 12. 0x7: The BCLK is obtained by dividing the MCLK by 16. 0x8: The BCLK is obtained by dividing the MCLK by 24. 0x9: The BCLK is obtained by dividing the MCLK by 32. 0xA: The BCLK is obtained by dividing the MCLK by 48. 0xB: The BCLK is obtained by dividing the MCLK by 64. Other values: The BCLK is obtained by dividing the MCLK by 8.	0x1



## RX\_IF\_ATTRI

RX\_IF\_ATTRI is an interface attribute configuration register for the RX channel.

Offset Address: 0x1000+0x100×n Total Reset Value: 0xE408\_0004

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0xE4
[23:20]	RW	rx_sd_source_sel	SD0, SD1, SD2, and SD3 source select 0x0: I2S TX0 0x1: I2S TX1 ... 0x7: I2S TX7 0x8: I2S RX0 0x9: I2S RX1 ... 0xF: I2S RX7	0x0
[19]	-	reserved	Reserved	0x1
[18:16]	RW	rx_trackmode	Audio-left and audio-right channel mode control in non-TDM I <sup>2</sup> S mode 000: The sound is not processed. 001: The sounds in two channels are audio-left channel sounds. 010: The sounds in two channels are audio-right channel sounds. 011: The sounds in two channels are exchanged. 100: The sounds of two channels are added and then output. 101: The audio-left channel is muted, and the sound of the audio-right channel is from the original audio-right channel. 110: The audio-right channel is muted, and the sound of the audio-left channel is from the original audio-left channel. 111: The audio-left and audio-right channels are muted. Note: <b>trackmode</b> is still valid in mono-audio channel TX mode.	0x0



Bits	Access	Name	Description	Reset
[15:8]	RW	rx_sd_offset	Number of BCLK clocks delayed for data relative to the frame sync signal in PCM mode  0x00: 0 bit clocks 0x01: 1 bit clocks 0x02: 2 bit clocks ... 0xFE: 254 bit clocks 0xFF: 255 bit clocks	0x00
[7]	RW	rx_multislot_en	TDM validity indicator  0: invalid (normal mode) 1: valid	0x0
[6]	-	reserved	Reserved	0x0
[5:4]	RW	rx_ch_num	Number of RX channels  rx_multislot_en = 0  00: 1-channel RX (SD0 data line) 01: 2-channel RX (SD0 data line) Other values: reserved  rx_multislot_en = 1  Number of RX channels in TDM mode. 00: 2-channel RX (SD0 data line) 01: 4-channel RX (SD0 data line) Other values: reserved Note: The data line SD0 must be used for multi-channel RX.	0x0
[3:2]	RW	rx_i2s_precision	Data sampling precision  I <sup>S</sup> mode 01: 16-bit 10: 24-bit Other values: reserved  PCM mode 01: 16-bit Other values: reserved	0x1
[1:0]	RW	rx_mode	Interface mode of the RX channel  00: I <sup>S</sup> mode	0x0



Bits	Access	Name	Description	Reset
			01: PCM mode Other values: reserved	

## RX\_DSP\_CTRL

RX\_DSP\_CTRL is an RX channel control register.

Offset Address: 0x1004+0x100×n Total Reset Value: 0x2000\_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	Reserved	0x0
[29]	RO	rx_disable_done	RX channel disable completion identifier 0: not complete 1: complete	0x1
[28]	RW	rx_enable	RX channel start/stop indicator bit 0: stop 1: start	0x0
[27]	RW	bypass_en	Operation bypass enable (the control function still takes effect). 0: disabled 1: enabled ( <b>trackmode</b> is bypassed.)	0x0
[26:0]	-	reserved	Reserved	0x00000000

## RX\_BUFF\_SADDR\_LOW

RX\_BUFF\_SADDR\_LOW is the DDR buffer start address register for the RX channel.

Offset Address: 0x1080+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	rx_buff_saddr_low	DDR buffer start address of the RX channel. Its unit is byte.	0x00000000

## RX\_BUFF\_SIZE

RX\_BUFF\_SIZE is a DDR buffer size register for the RX channel.



Offset Address: 0x1084+0x100×n Total Reset Value: 0x0000\_0032

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	rx_buff_size	DDR buffer size of the RX channel. Its unit is byte.  <b>NOTE</b> The buffer size must be an integral multiple of 32 bytes.	0x000032

## RX\_BUFF\_WPTR

RX\_BUFF\_WPTR is a DDR buffer write address register for the RX channel.

Offset Address: 0x1088+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	rx_buff_wptr	Write address for the DDR buffer of the RX channel. Its unit is byte.  <b>NOTE</b> <ul style="list-style-type: none"><li>④ The write address in the RX direction is maintained by the logic and is the offset address relative to the start address for the DDR buffer.</li><li>④ The write address must be 128x2-bit aligned. The lower 4 bits of the software are unavailable.</li></ul>	0x000000

## RX\_BUFF\_RPTR

RX\_BUFF\_RPTR is a DDR buffer write address register for the RX channel.

Offset Address: 0x108C+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	rx_buff_rptr	Write address for the DDR buffer of the RX channel. Its unit is byte.  <b>DNOTE</b>	0x000000



Bits	Access	Name	Description	Reset
			<ul style="list-style-type: none"><li>☞ The write address in the RX direction is maintained by the logic and is the offset address relative to the start address for the DDR buffer.</li><li>☞ The write address must be 32-byte aligned.</li></ul>	

## RX\_BUFF\_ALFULL\_TH

RX\_BUFF\_ALFULL\_TH is a DDR buffer almost full threshold register for the RX channel.

Offset Address: 0x1090+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	rx_buff_alfull_th	Almost full threshold for the DDR buffer of the RX channel. Its unit is byte. If the available space of the DDR buffer is below the almost full threshold, the almost full raw interrupt is generated.   <b>NOTE</b> If the rx_alfull_int interrupt is used, the field value must be an integral multiple of 16 bytes and greater than or equal to 0x40.	0x000000

## RX\_TRANS\_SIZE

RX\_TRANS\_SIZE is data transfer length register for the RX channel.

Offset Address: 0x1094+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	rx_trans_size	After the RX channel receives the audio data with the length of rx_trans_size (in byte), a transfer completion interrupt is generated.	0x000000



## RX\_WPTR\_TMP

RX\_WPTR\_TMP is the write address storage register for the RX channel upon reporting of the transfer completion interrupt.

Offset Address: 0x1098+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RO	rx_wptr_tmp	This field is used to save the write address for the RX channel when a transfer completion interrupt is reported.	0x000000

## RX\_INT\_ENA

RX\_INT\_ENA is an interrupt enable register for the RX channel.

Offset Address: 0x10A0+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7]	RW	rx_if_full_lost_int_ena	Raw interrupt enable for the data loss caused by the interface full data of the RX channel 0: disabled 1: enabled	0x0
[6]	-	reserved	Reserved	0x0
[5]	RW	rx_stop_int_ena	Stop interrupt enable for the RX channel 0: disabled 1: enabled	0x0
[4]	RW	rx_ififo_full_int_ena	Interface FIFO overflow interrupt enable for the RX channel 0: disabled 1: enabled	0x0
[3]	RW	rx_bfifo_full_int_ena	Bus FIFO overflow interrupt enable for the RX channel 0: disabled 1: enabled	0x0
[2]	RW	rx_alfull_int_ena	DDR buffer almost full interrupt enable	0x0



Bits	Access	Name	Description	Reset
			for the RX channel 0: disabled 1: enabled	
[1]	RW	rx_full_int_ena	DDR buffer full interrupt for the RX channel 0: disabled 1: enabled	0x0
[0]	RW	rx_trans_int_ena	Transfer completion interrupt enable for the RX channel 0: disabled 1: enabled	0x0

## RX\_INT\_RAW

**RX\_INT\_RAW** is a raw interrupt register for the RX channel.

Offset Address: 0x10A4+0x100×n Total Reset Value: 0x0000\_0020

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7]	RO	rx_if_full_lost_int_raw	Raw interrupt enable for the data loss caused by the interface full data of the RX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[6]	-	reserved	Reserved	0x0
[5]	RO	rx_stop_int_raw	Stop raw interrupt of the RX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x1
[4]	-	reserved	Reserved	0x0
[3]	RO	rx_fifo_full_int_raw	Bus FIFO overflow raw interrupt of the RX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[2]	RO	rx_alfull_int_raw	Raw interrupt enable for the data loss caused by the interface full data of the	0x0



Bits	Access	Name	Description	Reset
			RX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	
[1]	RO	rx_full_int_raw	DDR buffer full raw interrupt of the RX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[0]	RO	rx_trans_int_raw	Transfer completion raw interrupt of the RX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0

## RX\_INT\_STATUS

RX\_INT\_STATUS is an interrupt status register for the RX channel.

Offset Address: 0x10A8+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7]	RO	tx_if_full_lost_int_status	Interrupt status of the data loss caused by the interface full data of the RX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	-	reserved	Reserved	0x0
[5]	RO	rx_stop_int_status	Stop interrupt status of the RX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	rx_ififo_full_int_status	Interface FIFO overflow interrupt of the RX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	RO	rx_bfifo_full_int_status	Bus FIFO overflow interrupt status of the RX channel 0: No interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
			1: An interrupt is generated.	
[2]	RO	rx_alfull_int_status	DDR buffer almost full interrupt status of the RX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	rx_full_int_status	DDR buffer full interrupt status of the RX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	rx_trans_int_status	Transfer completion interrupt status of the RX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## RX\_INT\_CLR

RX\_INT\_CLR is an interrupt clear register for the RX channel.

Offset Address: 0x10AC+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7]	RW	rx_if_full_lost_int_clear	Data full loss interrupt clear for the RX channel 0: not cleared 1: cleared	0x0
[6]	-	reserved	Reserved	0x0
[5]	RW	rx_stop_int_clear	Stop interrupt clear for the RX channel 0: not cleared 1: cleared	0x0
[4]	RW	rx_ififo_full_int_clear	Interface FIFO overflow interrupt clear for the RX channel 0: not cleared 1: cleared	0x0
[3]	RW	rx_bfifo_full_int_clear	Bus FIFO overflow interrupt clear for the RX channel	0x0



Bits	Access	Name	Description	Reset
			0: not cleared 1: cleared	
[2]	RW	rx_alfull_int_clear	DDR buffer almost full interrupt clear for the RX channel 0: not cleared 1: cleared	0x0
[1]	RW	rx_full_int_clear	DDR buffer full interrupt clear for the RX channel 0: not cleared 1: cleared	0x0
[0]	WO	rx_trans_int_clear	Transfer completion interrupt clear for the RX channel 0: not cleared 1: cleared	0x0

## TX\_IF\_ATTRI

TX\_IF\_ATTRI is an interface attribute configuration register for the TX channel.

Offset Address: 0x2000+0x100×m Total Reset Value: 0xE400\_0004

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	Reserved	0x3
[29:28]	-	reserved	Reserved	0x2
[27:26]	-	reserved	Reserved	0x1
[25:24]	-	reserved	Reserved	0x0
[23:20]	RW	tx_sd_source_sel	SD0, SD1, SD2, and SD3 source select 0x0: I2S TX0 0x1: I2S TX1 ... 0x7: I2S TX7 0x8: I2S RX0 0x9: I2S RX1 ... 0xF: I2S RX7	0x0



Bits	Access	Name	Description	Reset
[19]	-	reserved	Reserved	0x0
[18:16]	RW	tx_trackmode	<p>Audio-left and audio-right channel mode control in I<sup>2</sup>S mode</p> <p>000: The sound is not processed.</p> <p>001: The sounds in two channels are audio-left channel sounds.</p> <p>010: The sounds in two channels are audio-right channel sounds.</p> <p>011: The sounds in two channels are exchanged.</p> <p>100: The sounds of two channels are added and then output.</p> <p>101: The audio-left channel is muted, and the sound of the audio-right channel is from the original audio-right channel.</p> <p>110: The audio-right channel is muted, and the sound of the audio-left channel is from the original audio-left channel.</p> <p>111: The audio-left and audio-right channels are muted.</p> <p>Note: <b>trackmode</b> is still valid in mono-audio channel TX mode.</p>	0x0
[15:8]	RW	tx_sd_offset	<p>Number of BCLK clocks delayed for data relative to the frame sync signal in PCM mode</p> <p>0x00: 0 bit clocks</p> <p>0x01: 1 bit clock</p> <p>0x02: 2 bit clocks</p> <p>...</p> <p>0xFE: 254 bit clocks</p> <p>0xFF: 255 bit clocks</p>	0x00
[7]	RO	reserved	Reserved	0x0
[6]	RO	reserved	Reserved	0x0
[5:4]	RW	tx_ch_num	<p>Number of TX channels</p> <p>00: 1-channel TX</p> <p>01: 2-channel TX</p>	0x0



Bits	Access	Name	Description	Reset
			Other values: reserved	
[3:2]	RW	tx_i2s_precision	Data sampling precision <ul style="list-style-type: none"><li>● I<sup>S</sup> normal mode:<ul style="list-style-type: none"><li>00: Reserved</li><li>01: 16-bit</li><li>10: 24-bit</li><li>11: Reserved</li></ul></li><li>● PCM normal mode:<ul style="list-style-type: none"><li>01: 16-bit</li></ul></li></ul> <p>Other values: reserved</p>	0x1
[1:0]	RW	tx_mode	Interface mode of the TX channel <ul style="list-style-type: none"><li>00: I<sup>S</sup> mode</li><li>01: PCM mode</li></ul> <p>Other values: reserved</p>	0x0

## TX\_DSP\_CTRL

TX\_DSP\_CTRL is a TX channel control register.

Offset Address: 0x2004+0x100×m Total Reset Value: 0x2000\_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	Reserved	0x0
[29]	RO	tx_disable_done	TX channel disable completion identifier <ul style="list-style-type: none"><li>0: not complete</li><li>1: complete</li></ul>	0x1
[28]	RW	tx_enable	TX channel start/stop indicator bit <ul style="list-style-type: none"><li>0: stop</li><li>1: start</li></ul>	0x0
[27]	RW	bypass_en	Operation bypass enable (the control function still takes effect). <ul style="list-style-type: none"><li>0: disabled</li><li>1: enabled (The functions such as volume control, track mode, and fade-in/fade-out are bypassed.)</li></ul>	0x0



Bits	Access	Name	Description	Reset
[26:24]	-	reserved	Reserved	0x0
[23:20]	RW	fade_out_rate	Fade-out rate 0x0: The fade-out rate changes every one sampling point. 0x1: The fade-out rate changes every two sampling points. 0x2: The fade-out rate changes every four sampling points. 0x3: The fade-out rate changes every eight sampling points. 0x4: The fade-out rate changes every 16 sampling points. 0x5: The fade-out rate changes every 32 sampling points. 0x6: The fade-out rate changes every 64 sampling points. 0x7: The fade-out rate changes every 128 sampling points. Other values: reserved	0x0
[19:16]	RW	fade_in_rate	Fade-in rate 0x0: The fade-in rate changes every one sampling point. 0x1: The fade-in rate changes every two sampling points. 0x2: The fade-in rate changes every four sampling points. 0x3: The fade-in rate changes every eight sampling points. 0x4: The fade-in rate changes every 16 sampling points. 0x5: The fade-in rate changes every 32 sampling points. 0x6: The fade-in rate changes every 64 sampling points. 0x7: The fade-in rate changes every 128 sampling points. Other values: reserved	0x0
[15]	-	reserved	Reserved	0x0



Bits	Access	Name	Description	Reset
[14:8]	RW	volume	Volume 0x00-0x28: muted 0x29: -80dB ... 0x7E: +5dB 0x7F: +6dB	0x00
[7:2]	-	reserved	Reserved	0x00
[1]	RW	mute_fade_en	Mute fade-in/fade-out control 0: disabled 1: enabled	0x0
[0]	RW	mute_en	Mute control 0: not muted 1: muted	0x0

## TX\_WS\_CNT

TX\_WS\_CNT is a WS cyclic count status register for the TX channel.

Offset Address: 0x2020+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RO	ws_count	FSCLK cyclic count register. Unit: FSCLK	0x000000

## TX\_BCLK\_CNT

TX\_BCLK\_CNT is a BCLK cyclic count status register for the TX channel.

Offset Address: 0x2024+0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RO	bclk_count	BCLK cyclic count register Unit: BCLK	0x000000



## TX\_BUFF\_SADDR\_LOW

TX\_BUFF\_SADDR\_LOW is the DDR buffer start address register for the TX channel.

Offset Address: 0x2080+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	tx_buff_saddr_low	Lower 32-bit DDR buffer start address register for the TX channel. Its unit is byte.   <b>NOTE</b> The start address of the DDR buffer must be 128 x 2 bits aligned.	0x00000000

## TX\_BUFF\_SIZE

TX\_BUFF\_SIZE is a DDR buffer size register for the TX channel.

Offset Address: 0x2084+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	tx_buff_size	DDR buffer size of the TX channel. Its unit is byte.  <b>DNOTE</b> The buffer size must be an integral multiple of 32 bytes.	0x000000

## TX\_BUFF\_WPTR

TX\_BUFF\_WPTR is a DDR buffer write address register for the TX channel.

Offset Address: 0x2088+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	tx_buff_wptr	Write address for the DDR buffer of the TX channel	0x000000



		<b>DNOTE</b> <ul style="list-style-type: none"><li>④ The write address in the TX direction is maintained by the logic and is the offset address relative to the start address for the DDR buffer.</li><li>④ The available space of the TX buffer must be greater than or equal to 32 bytes.</li><li>④ The write address is in the unit of byte on software and 32-byte aligned on hardware.</li></ul>	
--	--	---	--

## TX\_BUFF\_RPTR

TX\_BUFF\_RPTR is a DDR buffer read address register for the TX channel.

Offset Address: 0x208C+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	tx_buff_rptr	Read address for the DDR buffer of the TX channel <b>NOTE</b> <ul style="list-style-type: none"><li>④ The read address in the TX direction is maintained by the logic and is the offset address relative to the start address for the DDR buffer.</li><li>④ The address must be 128 x 2-bit aligned.</li></ul>	0x0000000

## TX\_BUFF\_ALEMPY\_TH

TX\_BUFF\_ALEMPY\_TH is a DDR buffer almost empty threshold register for the TX channel.

Offset Address: 0x2090+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	tx_buff_alempy_th	Almost empty threshold for the DDR buffer of the TX channel. Its unit is byte. If the available space of the DDR buffer is below the almost empty threshold, the almost empty raw interrupt is generated.	0x0000000



			<b>DNOTE</b> If the tx_alempty_int interrupt is used, the field value must be an integral multiple of 16 bytes and greater than or equal to 0x20.	
--	--	--	--	--

## TX\_TRANS\_SIZE

Offset Address: 0x2094+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	tx_trans_size	After the TX channel transmits the audio data with the length of tx_trans_size (in byte), a transfer completion interrupt is generated.	0x000000

## TX\_RPTR\_TMP

TX\_RPTR\_TMP is the read address storage register for the TX channel upon reporting of the transfer completion interrupt.

Offset Address: 0x2098+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RO	tx_rptr_tmp	This field is used to save the read address (in bytes) for the TX channel when the transfer completion interrupt is reported.	0x000000

## TX\_INT\_ENA

TX\_INT\_ENA is an interrupt enable register for the TX channel.

Offset Address: 0x20A0+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7]	RW	tx_dat_break_int_ena	Interface data break interrupt enable for the TX channel	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[6]	RW	tx_mfade_int_ena	Mute fade-in/fade-output completion interrupt enable for the TX channel 0: disabled 1: enabled	0x0
[5]	RW	tx_stop_int_ena	Stop interrupt enable for the TX channel 0: disabled 1: enabled	0x0
[4]	RW	tx_ififo_empty_int_ena	Interface FIFO underflow interrupt enable for the TX channel 0: disabled 1: enabled	0x0
[3]	RW	tx_bfifo_empty_int_ena	Bus FIFO underflow interrupt enable for the TX channel 0: disabled 1: enabled	0x0
[2]	RW	tx_alempty_int_ena	DDR buffer almost empty interrupt enable for the TX channel 0: disabled 1: enabled	0x0
[1]	RW	tx_empty_int_ena	DDR buffer empty interrupt for the TX channel 0: disabled 1: enabled	0x0
[0]	RW	tx_trans_int_ena	Transfer completion interrupt enable for the TX channel 0: disabled 1: enabled	0x0

## TX\_INT\_RAW

TX\_INT\_RAW is a raw interrupt register for the TX channel.

Offset Address: 0x20A4+0x100×m Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7]	RO	tx_dat_break_int_raw	Interface data break raw interrupt of the TX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[6]	RO	tx_mfade_int_raw	Mute fade-in/fade-output completion raw interrupt of the TX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[5]	RO	tx_stop_int_raw	Stop raw interrupt of the TX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[4]	RO	tx_ififo_empty_int_raw	Interface FIFO underflow raw interrupt of the TX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[3]	RO	tx_bfifo_empty_int_raw	Bus FIFO underflow raw interrupt of the TX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[2]	RO	tx_alempty_int_raw	DDR buffer almost empty raw interrupt of the TX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[1]	RO	tx_empty_int_raw	DDR buffer empty raw interrupt of the TX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0
[0]	RO	tx_trans_int_raw	Transfer completion raw interrupt of the TX channel 0: No raw interrupt is generated. 1: A raw interrupt is generated.	0x0



## TX\_INT\_STATUS

TX\_INT\_STATUS is an interrupt status register for the TX channel.

Offset Address: 0x20A8+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7]	RO	tx_dat_break_int_status	Status of the interface data break interrupt of the TX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	RO	tx_mfade_int_status	Status of the mute fade-in/fade-output completion interrupt of the TX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[5]	RO	tx_stop_int_status	Status of the stop interrupt of the TX interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	tx_ififo_empty_int_status	Status of the interface FIFO underflow interrupt of the TX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	RO	tx_bfifo_empty_int_status	Status of the bus FIFO underflow interrupt of the TX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	RO	tx_alempty_int_status	DDR buffer almost empty interrupt of the TX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	tx_empty_int_status	Status of the DDR buffer empty interrupt of the TX channel 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	tx_trans_int_status	Status of the transfer completion interrupt of the TX channel	0x0



Bits	Access	Name	Description	Reset
			0: No interrupt is generated. 1: An interrupt is generated.	

## TX\_INT\_CLR

TX\_INT\_CLR is an interrupt clear register for the TX channel.

Offset Address: 0x20AC+0x100×m Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7]	RW	tx_dat_break_int_clear	Interface data break interrupt clear for the TX channel 0: not cleared 1: cleared	0x0
[6]	RW	tx_mfade_int_clear	Mute fade-in/fade-output completion interrupt clear for the TX channel 0: not cleared 1: cleared	0x0
[5]	RW	tx_stop_int_clear	Stop interrupt clear for the TX channel 0: not cleared 1: cleared	0x0
[4]	RW	tx_ififo_empty_int_clear	Interface FIFO underflow interrupt clear for the TX channel 0: not cleared 1: cleared	0x0
[3]	RW	tx_bfifo_empty_int_clear	Bus FIFO underflow interrupt clear for the TX channel. 0: not cleared 1: cleared	0x0
[2]	RW	tx_alempty_int_clear	DDR buffer almost empty interrupt clear for the TX channel 0: not cleared 1: cleared	0x0
[1]	RW	tx_empty_int_clear	DDR buffer empty interrupt clear for the RX channel	0x0



Bits	Access	Name	Description	Reset
			0: not cleared 1: cleared	
[0]	RW	tx_trans_int_clear	Transfer completion interrupt clear for the TX channel 0: not cleared 1: cleared	0x0

## 11.2 Audio Codec

### 11.2.1 Overview

The chip is integrated with high-performance audio codecs, including high-quality playback DAC (90 dB DR A-weighted), one single-ended lineout, high-quality recording ADC (90 dB DR A-weighted), two differentiated single-ended inputs, and 0–30 dB MIC input. The gain control step is 2 dB. The boost gain is 20 dB. The PS data interface supports the 8 kHz–48 kHz standard sampling rates.

### 11.2.2 Features

The audio codec module has the following features:

- 90 dBA DR DAC, supporting one single-ended lineout.
- DAC digital volume control range: -121 dB to +6 dB, at 1 dB step
- 90 dBA DR ADC, supporting two differentiated singled-ended inputs
- Analog volume control range of ADC channel: 0 dB to 30 dB, at 2 dB step. The boost gain is 20 dB.
- Provides internal MIC biasing.
- Audio sampling frequencies: 48 kHz, 44.1 kHz, and 32 kHz

The sampling frequencies of each series are as follows:

- The 32 kHz sampling frequencies series include 8 kHz, 16 kHz, and 32 kHz.
- The 44.1 kHz sampling frequencies series include 11.025 kHz, 22.05 kHz, and 44.1 kHz.
- The 48 kHz sampling frequencies series include 12 kHz, 24 kHz, and 48 kHz.



## 11.2.3 Register Summary

Table 11-3 describes the AudioCodec registers.

**Table 11-3** Summary of AudioCodec registers (base address: 0x17C4\_0000)

Offset Address	Name	Description	Page
0x0000	AUDIO_ANA_CTRL_0	Audio codec analog control register 0	<a href="#">11-46</a>
0x0004	AUDIO_ANA_CTRL_1	Audio codec analog control register 1	<a href="#">11-49</a>
0x0008	AUDIO_ANA_CTRL_2	Audio codec analog control register 2	<a href="#">11-50</a>
0x000C	AUDIO_ANA_CTRL_3	Audio codec analog control register 3	<a href="#">11-50</a>
0x00CC	AUDIO_CTRL_REG_1	Audio codec DIG control register 0	<a href="#">11-51</a>
0x00D0	AUDIO_DAC_REG_0	Audio codec DIG control register 1	<a href="#">11-52</a>
0x00D4	AUDIO_DAC_REG_1	Audio codec DIG control register 2	<a href="#">11-53</a>
0x00D8	AUDIO_ADC_REG_0	Audio codec DIG control register 3	<a href="#">11-54</a>

## 11.2.4 Register Description

### AUDIO\_ANA\_CTRL\_0

AUDIO\_ANA\_CTRL\_0 is audio codec analog control register 0.

Offset Address: 0x0000 Total Reset Value: 0x3434\_DFFF

Bits	Access	Name	Description	Reset
[31:27]	RW	LINEIN_R_GAIN	Gain control for the right-channel input PGA  0x0: 0dB 0x1: 2dB  ...  0xF: 30dB	0x06
[26:24]	RW	LINEIN_R_SEL	Input channel select for the right-channel input PGA	0x4



Bits	Access	Name	Description	Reset
			000: MIC differential input 001: MIC_R input 010: MIC_L input 011: LINEIN_L input 100: LINEIN_R input 101: LINEIN differential input Other values: reserved	
[23:19]	RW	LINEIN_L_GAIN	Gain control for the left-channel input PGA  0x00: 0dB 0x01: 2dB ... 0x0F: 30dB	0x06
[18:16]	RW	LINEIN_L_SEL	Input channel select for the left-channel input PGA  000: MIC differential input 001: MIC_L input 010: MIC_R input 011: LINEIN_R input 100: LINEIN_L input 101: LINEIN differential input Other values: reserved	0x4
[15]	RW	PD_MICBIAS1	Power-on/off control for MICBIAS1 0: normal working 1: powered off	0x1
[14]	-	reserved	Reserved	0x1
[13]	RW	EN_LDO	Power-on/off control for LDO 0: powered off 1: powered on	0x0
[12]	RW	PD_DAC_VREF	Power-on/off control for DAC_VREF 0: normal working 1: powered off	0x1
[11]	RW	PD_CTCM_TX	Power-on/off control for the TX channel CTCM_BUFFER 0: normal working	0x1



Bits	Access	Name	Description	Reset
			1: powered off	
[10]	RW	PD_CTCM_RX	Power-on/off control for the RX channel CTCM_BUFFER 0: normal working 1: powered off	0x1
[9]	RW	PD_IBIAS	Power-on/off control for the bias current generation circuit 0: normal working 1: powered off	0x1
[8]	RW	PD_VREF	Power-on/off control for VREF 0: normal working 1: powered off	0x1
[7]	RW	PD_LINEOUTL	Power-on/off control for the left-channel LINEOUT 0: normal working 1: powered off	0x1
[6]	RW	PD_LINEOUTR	Power-on/off control for the right-channel LINEOUT 0: normal working 1: powered off	0x1
[5]	RW	PD_ADCL	Power-on/off control for the left-channel ADC 0: normal working 1: powered off	0x1
[4]	RW	PD_ADCR	Power-on/off control for the right-channel ADC 0: normal working 1: powered off	0x1
[3]	RW	PD_LINEIN_L	Power-on/off control for the left-channel input PGA 0: normal working 1: powered off	0x1
[2]	RW	PD_LINEIN_R	Power-on/off control for the right-channel input PGA 0: normal working	0x1



Bits	Access	Name	Description	Reset
			1: powered off	
[1]	RW	MUTE_LINEIN_L	Mute control for the left-channel input PGA 0: normal 1: muted	0x1
[0]	RW	MUTE_LINEIN_R	Mute control for the right-channel input PGA 0: normal 1: muted	0x1

## AUDIO\_ANA\_CTRL\_1

AUDIO\_ANA\_CTRL\_1 is audio codec analog control register 1.

Offset Address: 0x0004 Total Reset Value: 0xDF60\_5E65

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0xDF6
[19]	RW	MODE_ADCL	Mode select for the left-channel ADC 0: 6.144 MHz sampling rate (48 kHz series) 1: 4.096 MHz sampling rate (32 kHz series)	0x0
[18]	RW	MODE_ADCR	Mode select for the right-channel ADC 0: 6.144 MHz sampling rate (48 kHz series) 1: 4.096 MHz sampling rate (32 kHz series)	0x0
[17]	RW	BOOST_ADCL	Boost enable for the left-channel ADC 0: disabled 1: enabled (20 dB)	0x0
[16]	RW	BOOST_ADCR	Boost enable for the right-channel ADC 0: disabled 1: enabled (20 dB)	0x0
[15:2]	-	reserved	Reserved	0x1799
[1]	RW	EN_RCTUNE	ADC RC coefficient correction circuit	0x0



Bits	Access	Name	Description	Reset
			enable 0: disabled 1: normal working This register is a rising edge triggered signal. To enable the RC coefficient correction circuit to work, set this register to 0 and then to 1. After the system detects a rising edge, the RC correction circuit is triggered.	
[0]	RW	PD_RCTUNE	Power-on/off control for the ADC RC coefficient correction circuit 0: normal working 1: powered off	0x1

## AUDIO\_ANA\_CTRL\_2

AUDIO\_ANA\_CTRL\_2 is audio codec analog control register 2.

Offset Address: 0x0008 Total Reset Value: 0x0025\_5548

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	Reserved	0x0
[29:27]	RW	MICBIAS_ADJ	MICBIAS output voltage select 000: 2.1 V 001: 2.2 V ... 111: 2.8 V	0x0
[26:1]	-	reserved	Reserved	0x12AAA4
[0]	RW	ANA_LOOP	Analog loopback enable control 0: The DAC input is provided by the digital part. 1: The analog internal loopback is performed from the ADC to the DAC.	0x0

## AUDIO\_ANA\_CTRL\_3

AUDIO\_ANA\_CTRL\_3 is audio codec analog control register 3.



Offset Address: 0x000C Total Reset Value: 0x0429\_3B50

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x010A4E
[9]	RW	MUTE_DACL	Mute control for the left-channel DAC 0: normal 1: muted	0x1
[8]	RW	MUTE_DACR	Mute control for the right-channel DAC 0: normal 1: muted	0x1
[7:0]	-	reserved	Reserved	0x50

## AUDIO\_CTRL\_REG\_1

AUDIO\_CTRL\_REG\_1 is audio codec DIG control register 0.

Offset Address: 0x00CC Total Reset Value: 0x00F3\_5A4A

Bits	Access	Name	Description	Reset
[31]	RW	dacl_RST_N	DACL reset signal 0: reset 1: reset deasserted	0x0
[30]	RW	dacr_RST_N	DACR reset signal 0: reset 1: reset deasserted	0x0
[29]	RW	adcl_RST_N	ADCL reset signal 0: reset 1: reset deasserted	0x0
[28]	RW	adcr_RST_N	ADC reset signal 0: reset 1: reset deasserted	0x0
[27]	RW	dACL_en	DACL enable 0: disabled 1: enabled	0x0
[26]	RW	dACR_en	DACR enable 0: disabled	0x0



Bits	Access	Name	Description	Reset
			1: enabled	
[25]	RW	adcl_en	ADCL enable 0: disabled 1: enabled	0x0
[24]	RW	adcr_en	ADCR enable 0: disabled 1: enabled	0x0
[23:18]	RO	reserved	Reserved	0x3C
[17:13]	RW	i2s1_fs_sel	Sampling rate of the I <sup>S</sup> channel 0x18: mclk/512/2 0x19: mclk/256/2 0x1A: mclk/128/2 0x1B: mclk/64/2 0x1C~0x1F: mclk/32/2 Other values: reserved	0x1A
[12:0]	RO	reserved	Reserved	0x1A4A

## AUDIO\_DAC\_REG\_0

AUDIO\_DAC\_REG\_0 is audio codec DIG control register 1.

Offset Address: 0x00D0 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31]	RW	smutel	DACL soft mute control 0: disabled 1: enabled	0x0
[30]	RW	smuter	DACR soft mute control 0: disabled 1: enabled	0x0
[29]	RW	sunmutel	DACL soft unmute control 0: disabled 1: enabled	0x0
[28]	RW	sunmuter	DACR soft unmute control	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[27]	RW	dacvu	DAC volume update control 0: not updated 1: updated	0x0
[26:25]	RW	mutel_rate	DACL soft mute rate control 00: fs/2 01: fs/8 10: fs/32 11: fs/64	0x0
[24:23]	RW	muter_rate	DACR soft mute rate control 00: fs/2 01: fs/8 10: fs/32 11: fs/64	0x0
[22:0]	RO	reserved	Reserved	0x000001

## AUDIO\_DAC\_REG\_1

AUDIO\_DAC\_REG\_1 is audio codec DIG control register 2.

Offset Address: 0x00D4 Total Reset Value: 0x0606\_2424

Bits	Access	Name	Description	Reset
[31]	RW	dacl_mute	DACL digital mute control 0: normal working 1: muted	0x0
[30:24]	RW	dacl_vol	DACL digital volume control. The volume is calculated as follows: (6 - dacl_vol x 1) dB When dacl_vol is 0x7F, the DACL is muted. 0x00: 6 dB 0x01: 5 dB 0x02: 4 dB	0x06



Bits	Access	Name	Description	Reset
			... 0x7E: -120 dB 0x7F: muted	
[23]	RW	dacr_mute	DACR digital mute control 0: normal working 1: muted	0x0
[22:16]	RW	dacr_vol	DACR digital volume control The volume is calculated as follows: (6 – dacr_vol x 1) dB When dacr_vol is 0x7F, the DACR is muted. 0x00: 6dB 0x01: 5dB 0x02: 4dB ... 0x7E: -120dB 0x7F: mute	0x06
[15:0]	RO	reserved	Reserved	0x2424

## AUDIO\_ADC\_REG\_0

AUDIO\_ADC\_REG\_0 is audio codec DIG control register 3.

Offset Address: 0x00D8 Total Reset Value: 0x1E1E\_0001

Bits	Access	Name	Description	Reset
[31]	RW	adcl_mute	ADCL digital mute control 0: unmuted 1: muted	0x0
[30:24]	RW	adcl_vol	ADCL volume control bits. The volume is calculated as follows: (30 – adcl_vol x 1) dB 0x00: 30 dB 0x01: 29 dB 0x02: 28 dB ...	0x1E



Bits	Access	Name	Description	Reset
			0x7E: -96 dB 0x7F: -97 dB	
[23]	RW	adcr_mute	ADCR digital mute control 0: unmuted 1: muted	0x0
[22:16]	RW	adcr_vol	ADCR volume control The volume is calculated as follows: (30 – adcr_vol x 1) dB  0x00: 30 dB 0x01: 29 dB 0x02: 28 dB 0x...  0x7E: -96 dB 0x7F: -97 dB	0x1E
[15]	RW	adcl_hpf_en	ADCL high-pass filter enable 0: disabled 1: enabled	0x0
[14]	RW	adcr_hpf_en	ADCR high-pass filter enable 0: disabled 1: enabled	0x0
[13:0]	-	reserved	Reserved	0x0001



# Contents

<b>12 Peripherals.....</b>	<b>12-1</b>
12.1 I <sub>C</sub> Controller .....	12-1
12.1.1 Overview .....	12-1
12.1.2 Function Description.....	12-1
12.1.3 Functional Block Diagram .....	12-2
12.1.4 Timing Description Principles.....	12-2
12.1.5 Operating Mode.....	12-11
12.1.6 Register Summary .....	12-15
12.1.7 Register Description.....	12-16
12.2 UART.....	12-30
12.2.1 Overview .....	12-30
12.2.2 Features.....	12-31
12.2.3 Function Description.....	12-31
12.2.4 Operating Mode.....	12-33
12.2.5 Register Summary .....	12-36
12.2.6 Register Description.....	12-36
12.3 eMMC/SDIO/SD Card Controller.....	12-49
12.3.1 Overview .....	12-49
12.3.2 Features.....	12-49
12.3.3 Function Description.....	12-50
12.3.4 Register Summary .....	12-61
12.3.5 Register Description.....	12-63
12.4 SPI.....	12-93
12.4.1 Overview .....	12-93
12.4.2 Features.....	12-93
12.4.3 Function Description.....	12-94
12.4.4 Peripheral Bus Timings .....	12-95
12.4.5 Operating Mode.....	12-102
12.4.6 Register Summary .....	12-105
12.4.7 Register Description.....	12-106



12.5 GPIO.....	12-114
12.5.1 Overview .....	12-114
12.5.2 Features.....	12-114
12.5.3 Operating Mode.....	12-115
12.5.4 Register Summary .....	12-116
12.5.5 Register Description.....	12-117
12.6 USB DRD.....	12-121
12.6.1 Overview .....	12-121
12.6.2 Function Description.....	12-122
12.6.3 Operating Mode.....	12-124
12.6.4 Eye Pattern Parameters of USB 2.0 PHY .....	12-124
12.6.5 Register Summary .....	12-124
12.6.6 Register Description.....	12-125
12.7 LSADC.....	12-126
12.7.1 Overview .....	12-126
12.7.2 Features.....	12-127
12.7.3 Operating Mode.....	12-127
12.7.4 Register Summary .....	12-130
12.7.5 Register Description.....	12-131
12.8 PWM0.....	12-141
12.8.1 Overview .....	12-141
12.8.2 Configuration Process .....	12-141
12.8.3 Summary of PWM0 Registers .....	12-142
12.8.4 Description of PWM0 Registers .....	12-143
12.9 PWM1.....	12-145
12.9.1 Overview .....	12-145
12.9.2 Features.....	12-145
12.9.3 Operating Mode.....	12-146
12.9.4 Configuration Process .....	12-149
12.9.5 Summary of Variables in the Register Offset Addresses .....	12-151
12.9.6 Register Summary .....	12-152
12.9.7 Register Description.....	12-153



## Figures

<b>Figure 12-1</b> Functional block diagram of the controller.....	12-2
<b>Figure 12-2</b> 7-bit addressing, write timing.....	12-5
<b>Figure 12-3</b> 7-bit addressing, direct read timing .....	12-6
<b>Figure 12-4</b> 10-bit addressing, write timing .....	12-7
<b>Figure 12-5</b> 10-bit addressing, combined read timing .....	12-8
<b>Figure 12-6</b> Non-standard timing.....	12-10
<b>Figure 12-7</b> Typical application block diagram of the UART .....	12-32
<b>Figure 12-8</b> Frame format of the UART.....	12-32
<b>Figure 12-9</b> Functional block diagram of the MMC controller.....	12-51
<b>Figure 12-10</b> Typical application of the MMC controller.....	12-52
<b>Figure 12-11</b> Format of an MMC command .....	12-52
<b>Figure 12-12</b> Format of the response to an MMC command.....	12-53
<b>Figure 12-13</b> Operations by running an MMC non-data transfer command .....	12-53
<b>Figure 12-14</b> Single-block and multi-block read operations.....	12-55
<b>Figure 12-15</b> Single-block and multi-block write operations.....	12-55
<b>Figure 12-16</b> Data transfer format in 1-bit mode.....	12-56
<b>Figure 12-17</b> Data transfer format in 4-bit mode.....	12-56
<b>Figure 12-18</b> Data transfer format in 8-bit mode.....	12-57
<b>Figure 12-33</b> Logical block diagram of the USB 2.0 DRD module .....	12-122
<b>Figure 12-34</b> Single scanning procedure .....	12-127
<b>Figure 12-35</b> Take enabling channel 0 and channel 1 as an example, channel polling in continuous scanning mode.....	12-128
<b>Figure 12-36</b> Continuous scanning procedure .....	12-128
<b>Figure 12-37</b> Process of the 16-time average algorithm.....	12-130
<b>Figure 12-38</b> Left-aligned output waveform.....	12-146



<b>Figure 12-39</b> Right-aligned output waveform .....	12-147
<b>Figure 12-40</b> Center-aligned output waveform .....	12-147
<b>Figure 12-41</b> Output waveforms in sync mode.....	12-148



# Tables

<b>Table 12-1</b> Timing commands .....	12-3
<b>Table 12-2</b> 7-bit addressing, write timing description configurations .....	12-5
<b>Table 12-3</b> 7-bit addressing, direct read timing description configurations.....	12-6
<b>Table 12-4</b> 10-bit addressing, write timing description configurations.....	12-7
<b>Table 12-5</b> 10-bit addressing, combined read timing description configurations .....	12-8
<b>Table 12-6</b> Non-standard timing description configurations .....	12-10
<b>Table 12-7</b> Variables in the I <sub>2</sub> C register offset addresses .....	12-15
<b>Table 12-8</b> Summary of I <sub>2</sub> C registers.....	12-15
<b>Table 12-9</b> Summary of UART registers.....	12-36
<b>Table 12-10</b> Transfer modes.....	12-57
<b>Table 12-11</b> Transmission modes .....	12-57
<b>Table 12-12</b> Transmission modes .....	12-58
<b>Table 12-13</b> Summary of MMC registers (SDIO0/SDIO1 base address: 0x1003_0000/0x1004_0000) .....	12-61
<b>Table 12-15</b> Base addresses for GPIO registers.....	12-116
<b>Table 12-16</b> Summary of GPIO registers .....	12-117
<b>Table 12-17</b> Summary of USB PHY registers (base address: 0x1031_0000) .....	12-124
<b>Table 12-18</b> Summary of LSADC registers (base address: 0x1110_0000) .....	12-130
<b>Table 12-19</b> Summary of PWM0 registers (base address: 0x1102_9000) .....	12-142
<b>Table 12-20</b> Variable in the register offset address.....	12-152
<b>Table 12-21</b> Summary of PWM registers.....	12-152



# 12 Peripherals

## 12.1 I<sup>2</sup>C Controller

### 12.1.1 Overview

The inter-integrated circuit (I<sup>2</sup>C) controller provides the master I<sup>2</sup>C interface which allows the CPU to read/write data from/to slave devices on the I<sup>2</sup>C bus.

The SoC provides three I<sup>2</sup>C controllers, whose working clocks are determined by **i2c\_cksel** of the corresponding CRG register.

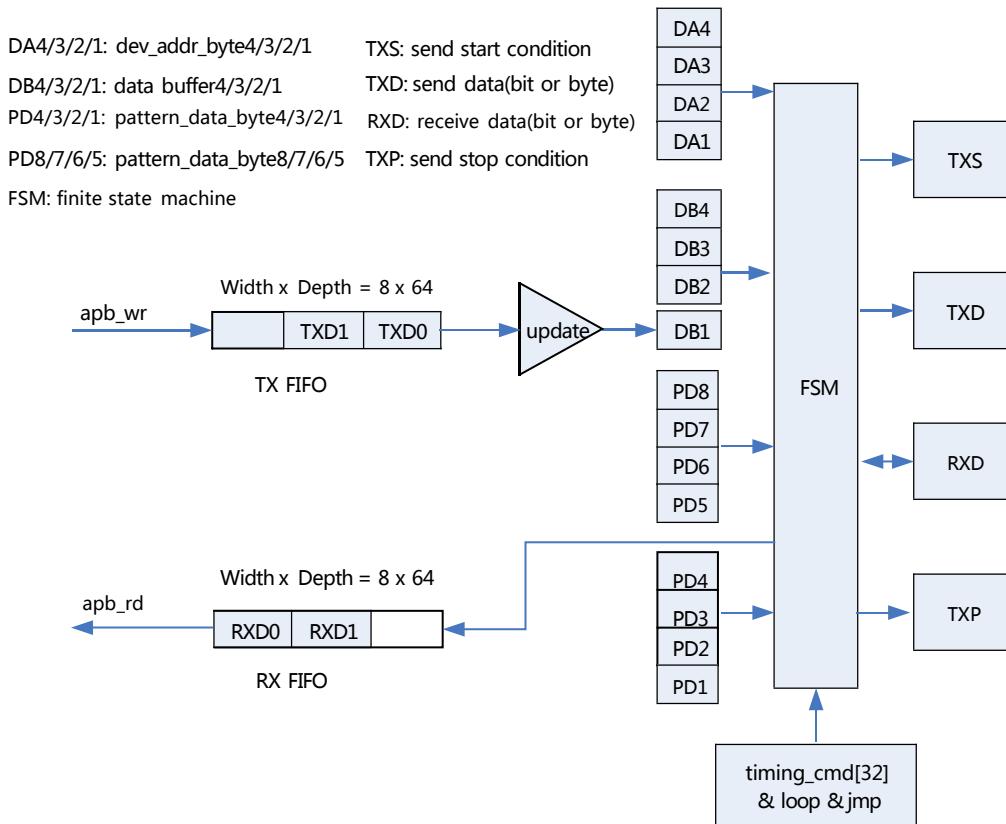
### 12.1.2 Function Description

The I<sup>2</sup>C controller has the following features:

- Master I<sup>2</sup>C interface which supports standard timing and non-standard timing
- Bus arbitration in the case of multiple master devices
- Clock synchronization and bit and byte waiting
- 7-bit standard address and 10-bit extended address
- Standard mode (100 kbit/s) and high-speed mode (400 kbit/s)
- General call and start byte
- CBUS components not supported
- DMA operation
- 64 x 8 bits TX FIFOs and 64 x 8 bits RX FIFOs

## 12.1.3 Functional Block Diagram

Figure 12-1 Functional block diagram of the controller



## 12.1.4 Timing Description Principles

Timing description includes timing format description and timing data preparation. Timing format description is implemented by configuring `I2C_TIMING_CMD`, `I2C_LOOP1`, `I2C_DST1`, `I2C_LOOP2`, `I2C_DST2`, `I2C_LOOP3`, and `I2C_DST3`. Both standard timings and non-standard timings can be configured.

- Timing data preparation is implemented by configuring `I2C_DEV_ADDR`, `I2C_DATA_BUF`, `I2C_PATTERN_DATA1`, `I2C_PATTERN_DATA2`, and `I2C_TX_FIFO`. The configured values need to be transmitted to the slave device.
- In timing format description, `I2C_TIMING_CMD` configures the timing command information, while `I2C_LOOP1`, `I2C_DST1`, `I2C_LOOP2`, `I2C_DST2`, `I2C_LOOP3`, and `I2C_DST3` configure the jump control information. Table 12-1 describes the timing command information that can be configured in `I2C_TIMING_CMD`.



**Table 12-1** Timing commands

Timing Command ID	Timing Command	Description
0x00	EXIT	Exit. This command is used for the controller to exit.
0x01	S	Bus START
0x02	SDA4	Send dev_addr_byte4.
0x03	SDA3	Send dev_addr_byte3.
0x04	SDA2	Send dev_addr_byte2.
0x05	SDA1	Send dev_addr_byte1.
0x06	SDB4	Send data_buf_byte4.
0x07	SDB3	Send data_buf_byte3.
0x08	SDB2	Send data_buf_byte2.
0x09	SDB1	Send data_buf_byte1.
0x0A	SPD8	Send pattern_data_byte8.
0x0B	SPD7	Send pattern_data_byte7.
0x0C	SPD6	Send pattern_data_byte6.
0x0D	SPD5	Send pattern_data_byte5.
0x0E	SPD4	Send pattern_data_byte4.
0x0F	SPD3	Send pattern_data_byte3.
0x10	SPD2	Send pattern_data_byte2.
0x11	SPD1	Send pattern_data_byte1.
0x12	RD	Receive 1 byte data. Note: When the RD command is executed, if the RX FIFO is full, the controller waits until the RX FIFO has spare space. During waiting, the controller does not change the status of the I <sup>C</sup> bus.
0x13	RACK	Receive low-level acknowledgement.
0x14	RNACK	Receive high-level non-acknowledgement.
0x15	RNC	Receive acknowledgement, regardless of the level.
0x16	SACK	Send low-level acknowledgement.



Timing Command ID	Timing Command	Description
0x17	SNACK	Send high-level non-acknowledgement.
0x18	JMPN1	Jump for limited times. The destination is specified by the DST1 register, and the number of jump times is specified by the LOOP1 register.
0x19	JMPN2	Jump for limited times. The destination is specified by the DST2 register, and the number of jump times is specified by the LOOP2 register.
0x1A	JMPN3	Jump for limited times. The destination is specified by the DST3 register, and the number of jump times is specified by the LOOP3 register.
0x1B	UNDEF	Undefined
0x1C	UNDEF	Undefined
0x1D	UDB1	Update data to <code>data_buf_byte1</code> from the TX FIFO.  Note: When the UDB1 command is executed, if the TX FIFO is empty, the controller waits until the TX FIFO has data. During waiting, the controller does not change the status of the I <sup>2</sup> C bus.
0x1E	SR	Bus repeated START
0x1F	P	Bus STOP

## Standard Timing - 7-Bit Addressing, Write Operation

Figure 12-2 7-bit addressing, write timing

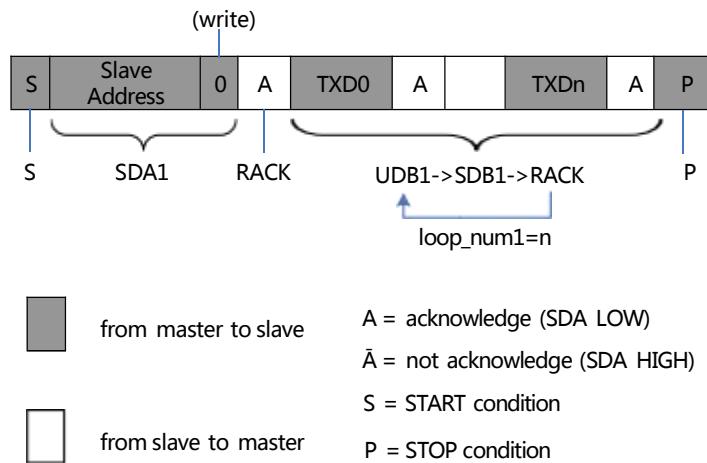


Table 12-2 describes the timing description configurations in Figure 12-2.

Table 12-2 7-bit addressing, write timing description configurations

Type	Register	Configuration
Timing format description	I2C_TIMING_CMD[0]	S
	I2C_TIMING_CMD[1]	SDA1
	I2C_TIMING_CMD[2]	RACK
	I2C_TIMING_CMD[3]	UDB1
	I2C_TIMING_CMD[4]	SDB1
	I2C_TIMING_CMD[5]	RACK
	I2C_TIMING_CMD[6]	JMPN1
	I2C_TIMING_CMD[7]	P
	I2C_TIMING_CMD[8]	EXIT
	I2C_LOOP1	$n$
Timing data preparation	I2C_DST1	3
	I2C_DEV_ADDR	Slave address+'0', a total of 8 bits
	I2C_TX_FIFO	Write TXD0, TXD1, ..., TXDn in turn.

## Standard Timing - 7-Bit Addressing, Direct Read Operation

Figure 12-3 7-bit addressing, direct read timing

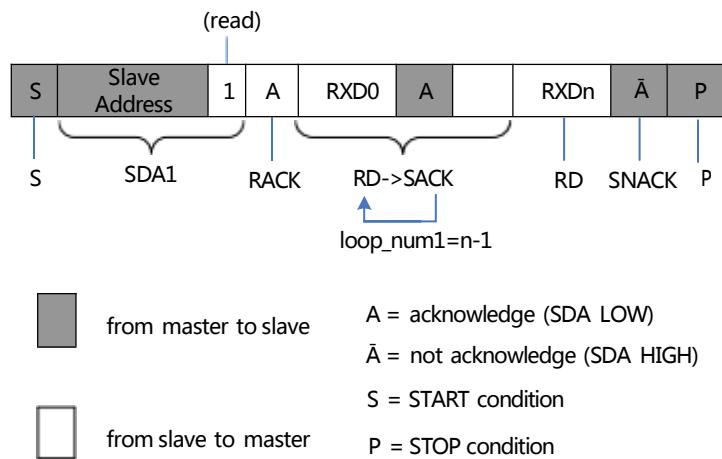


Table 12-3 describes the timing description configurations in Figure 12-3.

Table 12-3 7-bit addressing, direct read timing description configurations

Type	Register	Configuration
Timing format description	I2C_TIMING_CMD[0]	S
	I2C_TIMING_CMD[1]	SDA1
	I2C_TIMING_CMD[2]	RACK
	I2C_TIMING_CMD[3]	RD
	I2C_TIMING_CMD[4]	SACK
	I2C_TIMING_CMD[5]	JMPN1
	I2C_TIMING_CMD[6]	RD
	I2C_TIMING_CMD[7]	SNACK
	I2C_TIMING_CMD[8]	P
	I2C_TIMING_CMD[9]	EXIT
Timing data preparation	I2C_LOOP1	n-1
	I2C_DST1	3
Timing data preparation	I2C_DEV_ADDR	Slave address+'1', a total of 8 bits

 NOTE

Write the read-back data RXD0, RXD1, ..., and RXD<sub>n</sub> to I2C\_RX\_FIFO in turn.

## Standard Timing - 10-bit Addressing, Write Operation

**Figure 12-4** 10-bit addressing, write timing

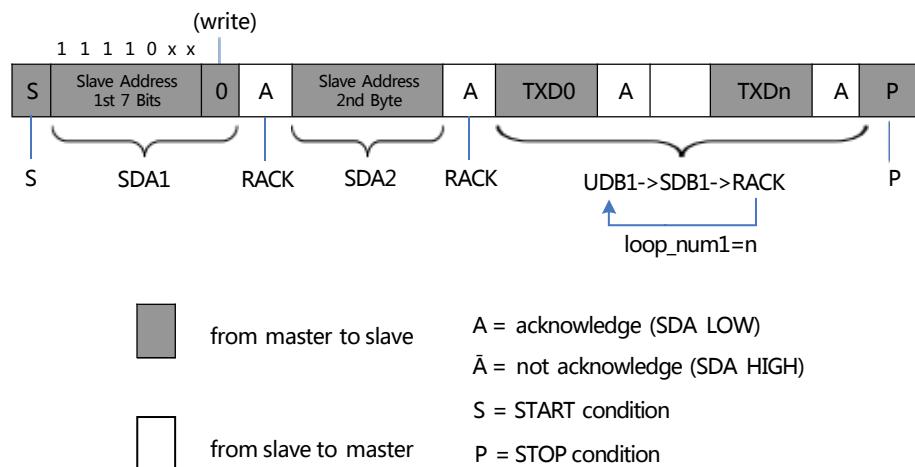


Table 12-4 describes the timing description configurations in Figure 12-4.

**Table 12-4** 10-bit addressing, write timing description configurations

Type	Register	Configuration
Timing format description	I2C_TIMING_CMD[0]	S
	I2C_TIMING_CMD[1]	SDA1
	I2C_TIMING_CMD[2]	RACK
	I2C_TIMING_CMD[3]	SDA2
	I2C_TIMING_CMD[4]	RACK
	I2C_TIMING_CMD[5]	UDB1
	I2C_TIMING_CMD[6]	WDB1
	I2C_TIMING_CMD[7]	RACK
	I2C_TIMING_CMD[8]	JMPN1
	I2C_TIMING_CMD[9]	P
	I2C_TIMING_CMD[10]	EXIT
	I2C_LOOP1	n

Type	Register	Configuration
	I2C_DST1	5
Timing data preparation	I2C_DEV_ADDR	Write 8 bits (first 7 bits of the slave address+'0') to dev_addr_byte1. Write (second byte of the slave address) to dev_addr_byte2.
	I2C_TX_FIFO	Write TXD0, TXD1, ..., TXDn in turn.

## Standard Timing - 10- Bit Addressing, Combined Read Operation

Figure 12-5 10-bit addressing, combined read timing

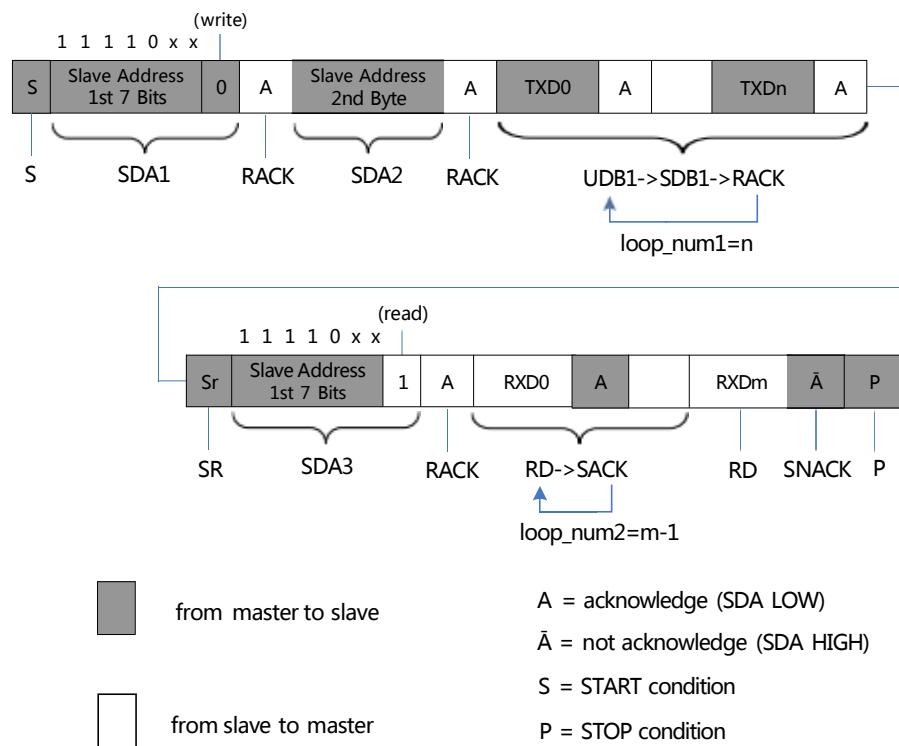


Table 12-5 describes the timing description configurations in Figure 12-5.

Table 12-5 10-bit addressing, combined read timing description configurations

Type	Register	Configuration
Timing format description	I2C_TIMING_CMD[0]	S
	I2C_TIMING_CMD[1]	SDA1



Type	Register	Configuration
	I2C_TIMING_CMD[2]	RACK
	I2C_TIMING_CMD[3]	SDA2
	I2C_TIMING_CMD[4]	RACK
	I2C_TIMING_CMD[5]	UDB1
	I2C_TIMING_CMD[6]	WDB1
	I2C_TIMING_CMD[7]	RACK
	I2C_TIMING_CMD[8]	JMPN1
	I2C_TIMING_CMD[9]	SR
	I2C_TIMING_CMD[10]	SDA3
	I2C_TIMING_CMD[11]	RACK
	I2C_TIMING_CMD[12]	RD
	I2C_TIMING_CMD[13]	SACK
	I2C_TIMING_CMD[14]	JMPN2
	I2C_TIMING_CMD[15]	RD
	I2C_TIMING_CMD[16]	SNACK
	I2C_TIMING_CMD[17]	P
	I2C_TIMING_CMD[18]	EXIT
	I2C_LOOP1	$n$
	I2C_DST1	5
	I2C_LOOP2	$m-1$
	I2C_DST2	12
Timing data preparation	I2C_DEV_ADDR	Write the first 7 bits of the slave address+'0' to dev_addr_byte1 Write the second byte of the slave address to dev_addr_byte2. Write the first 7 bits of the slave address+'1' to dev_addr_byte3.
	I2C_TX_FIFO	Write TXD0, TXD1, ..., TXDn in turn.

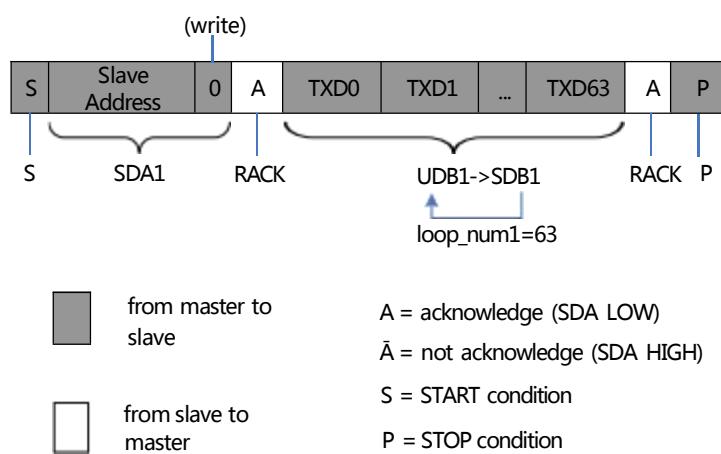
**NOTE**

Write the read-back data RXD0, RXD1, ..., and RXD $m$  to I2C\_RX\_FIFO in turn.

## Non-Standard Timing

The non-standard timing is the non-standard I<sup>2</sup>C protocol. [Figure 12-6](#) shows a non-standard timing. In this timing, an acknowledgement is received only after 64 bytes data is transmitted continuously to the slave device.

**Figure 12-6** Non-standard timing



[Table 12-6](#) describes the timing description configurations in [Figure 12-6](#).

**Table 12-6** Non-standard timing description configurations

Type	Register	Configuration
Timing format description	I2C_TIMING_CMD[0]	S
	I2C_TIMING_CMD[1]	SDA1
	I2C_TIMING_CMD[2]	RACK
	I2C_TIMING_CMD[3]	UDB1
	I2C_TIMING_CMD[4]	SDB1
	I2C_TIMING_CMD[5]	JMPN1
	I2C_TIMING_CMD[6]	RACK
	I2C_TIMING_CMD[7]	P
	I2C_TIMING_CMD[8]	EXIT
	I2C_LOOP1	63



Type	Register	Configuration
	I2C_DST1	3
Timing data preparation	I2C_DEV_ADDR	(Slave address+'0'), a total of 8 bits
	I2C_TX_FIFO	TXD0, TXD1, ..., and TXD63 in turn

## 12.1.5 Operating Mode

### 12.1.5.1 Data Transmission in Non-DMA Mode (Write Operation in Interrupt Mode)

Perform the following steps:

- Step 1** Configure [I2C\\_HCNT](#) and [I2C\\_LCNT](#) to enable the controller to work in the correct mode.
- Step 2** Set [I2C\\_TX\\_WATERMARK](#) to 0x20.
- Step 3** Configure [I2C\\_GLB](#) to enable the controller.
- Step 4** Complete timing configuration based on the actual timing by referring to section [12.1.4 "Timing Description Principles."](#)
- Step 5** Set [I2C\\_CTRL1](#) to 0x1 to start the controller.
- Step 6** Set [I2C\\_INTR\\_EN](#) to 0x811.
- Step 7** Query [I2C\\_INTR\\_STAT](#) in the interrupt service program. If [I2C\\_INTR\\_STAT\[arb\\_lost\]](#) or [I2C\\_INTR\\_STAT\[ack\\_bit\\_unmatch\]](#) is 1, an exception occurs; otherwise, if [I2C\\_INTR\\_STAT\[tx\\_lt\\_watermark\]](#) is 1, you can write up to 32 bytes data to be written to the TX FIFO.
  - If there still is data to be written, set [I2C\\_INTR\\_RAW](#) to 0x10 to clear the tx\_lt\_watermark interrupt that triggers this interrupt handling process and wait until the next time the interrupt service program is triggered.
  - If there is no data to be written, set [I2C\\_INTR\\_EN](#) to 0x1801 and wait for the all\_cmd\_done interrupt. When the all\_cmd\_done interrupt is received, all data has been written to the slave device.
- Step 8** Set [I2C\\_INTR\\_EN](#) to 0x0 and [I2C\\_INTR\\_RAW](#) to 0x1fff to clear the interrupt and disable interrupt reporting, making preparation for switchover between operating modes.

----End



### 12.1.5.2 Data Transmission in Non-DMA Mode (Write Operation in Query Mode)

Perform the following steps:

- Step 1** Configure [I2C\\_HCNT](#) and [I2C\\_LCNT](#) to enable the controller to work in the correct mode.
- Step 2** Configure [I2C\\_GLB](#) to enable the controller.
- Step 3** Complete timing configuration based on the actual timing by referring to section [12.1.4 "Timing Description Principles."](#)
- Step 4** Set [I2C\\_CTRL1](#) to 0x1 to start the controller.
- Step 5** Constantly query [I2C\\_FIFO\\_STAT\[tx\\_fifo\\_not\\_full\]](#) until it is 1 and write 1 byte data to be written to the TX FIFO. When all data to be written has been written to the TX FIFO, go to the next step.
- Step 6** Constantly query [I2C\\_INTR\\_RAW](#). If [I2C\\_INTR\\_RAW\[arb\\_lost\\_raw\]](#) or [I2C\\_INTR\\_RAW\[ack\\_bit\\_unmatch\\_raw\]](#) is 1, an exception occurs. If [I2C\\_INTR\\_RAW\[all\\_cmd\\_done\\_raw\]](#) is 1, all timings are complete.
- Step 7** Set [I2C\\_INTR\\_EN](#) to 0x0 and [I2C\\_INTR\\_RAW](#) to **0x1fff** to clear the interrupt and disable interrupt reporting, making preparation for switchover between operating modes.

----End

### 12.1.5.3 Data Transmission in Non-DMA Mode (Read Operation in Interrupt Mode)

Perform the following steps:

- Step 1** Configure [I2C\\_HCNT](#) and [I2C\\_LCNT](#) to enable the controller to work in the correct mode.
- Step 2** Set [I2C\\_RX\\_WATERMARK](#) to **0x20**.
- Step 3** Configure [I2C\\_GLB](#) to enable the controller.
- Step 4** Complete timing configuration based on the actual timing by referring to section [12.1.4 "Timing Description Principles."](#)
- Step 5** Set [I2C\\_CTRL1](#) to 0x1 to start the controller.
- Step 6** Set [I2C\\_INTR\\_EN](#) to **0x1805**.
- Step 7** Query [I2C\\_INTR\\_STAT](#) in the interrupt service program. If [I2C\\_INTR\\_STAT\[arb\\_lost\]](#) or [I2C\\_INTR\\_STAT\[ack\\_bit\\_unmatch\]](#) is 1, an exception occurs. If no exception occurs, and [I2C\\_INTR\\_STAT\[all\\_cmd\\_done\]](#) is 1, all data has been read back. If no exception occurs and [I2C\\_INTR\\_STAT\[all\\_cmd\\_done\]](#) is 0, the rx\_gt\_watermark interrupt is generated, and you can read the 32 bytes read-back data from the RX



FIFO. After reading the 32 bytes data, set `I2C_INTR_RAW` to 0x4 to clear the rx\_gt\_watermark interrupt that triggers this interrupt handling process and wait until the next time the interrupt service program is triggered.

- Step 8** Set `I2C_INTR_EN` to 0x0 and `I2C_INTR_RAW` to 0x1fff to clear the interrupt and disable interrupt reporting, making preparation for switchover between operating modes.
- End

#### 12.1.5.4 Data Transmission in Non-DMA Mode (Read Operation in Query Mode)

Perform the following steps:

- Step 1** Configure `I2C_HCNT` and `I2C_LCNT` to enable the controller to work in the correct mode.
- Step 2** Configure `I2C_GLB` to enable the controller.
- Step 3** Complete timing configuration based on the actual timing by referring to section [12.1.4 "Timing Description Principles."](#)
- Step 4** Set `I2C_CTRL1` to 0x1 to start the controller.
- Step 5** Constantly query `I2C_FIFO_STAT[rx_fifo_not_empty]` until it is 1 and read 1 byte data from the RX FIFO. When all data is read back, go to the next step.
- Step 6** Constantly query `I2C_INTR_RAW`. If `I2C_INTR_RAW[arb_lost_raw]` or `I2C_INTR_RAW[ack_bit_unmatch_raw]` is 1, an exception occurs. If `I2C_INTR_RAW[all_cmd_done_raw]` is 1, all timings are complete.
- Step 7** Set `I2C_INTR_EN` to 0x0 and `I2C_INTR_RAW` to 0x1fff to clear the interrupt and disable interrupt reporting, making preparation for switchover between operating modes.
- End

#### 12.1.5.5 Data Transmission in DMA Mode (Write Operation)

Perform the following steps:

- Step 1** Obtain a DMAC channel.
- Step 2** Configure and then start the DMAC channel.
- Step 3** Configure `I2C_HCNT` and `I2C_LCNT` to enable the controller to work in the correct mode.
- Step 4** Set `I2C_TX_WATERMARK` to 0x10. (The value here is only an example. The value should be configured based on actual scenarios.)
- Step 5** Configure `I2C_GLB` to enable the controller.



- Step 6** Complete timing configuration based on the actual timing by referring to section [12.1.4 "Timing Description Principles."](#)
- Step 7** Set `I2C_CTRL1` to 0x201 to start the controller.
- Step 8** Wait for the DMAC completion interrupt.
- Step 9** Constantly query `I2C_INTR_RAW`. If `I2C_INTR_RAW[arb_lost_raw]` or `I2C_INTR_RAW[ack_bit_unmatch_raw]` is 1, an exception occurs. If `I2C_INTR_RAW[all_cmd_done_raw]` is 1, all timings are complete.
- Step 10** Set `I2C_INTR_EN` to 0x0 and `I2C_INTR_RAW` to 0x1fff to clear the interrupt and disable interrupt reporting, making preparation for switchover between operating modes.
- End

### 12.1.5.6 Data Transmission in DMA Mode (Read Operation)

Perform the following steps:

- Step 1** Obtain a DMAC channel.
- Step 2** Configure and then start the DMAC channel.
- Step 3** Configure `I2C_HCNT` and `I2C_LCNT` to enable the controller to work in the correct mode.
- Step 4** Set `I2C_TX_WATERMARK` to 0x10. (The value here is only an example. The value should be configured based on actual scenarios.)
- Step 5** Configure `I2C_GLB` to enable the controller.
- Step 6** Complete timing configuration based on the actual timing by referring to section [12.1.4 "Timing Description Principles."](#)
- Step 7** Set `I2C_CTRL1` to 0x301 to start the controller.
- Step 8** Wait for the DMAC completion interrupt.
- Step 9** Constantly query `I2C_INTR_RAW`. If `I2C_INTR_RAW[arb_lost_raw]` or `I2C_INTR_RAW[ack_bit_unmatch_raw]` is 1, an exception occurs. If `I2C_INTR_RAW[all_cmd_done_raw]` is 1, all timings are complete.
- Step 10** Set `I2C_INTR_EN` to 0x0 and `I2C_INTR_RAW` to 0x1fff to clear the interrupt and disable interrupt reporting, making preparation for switchover between operating modes.
- End

### 12.1.5.7 Exception Handling Process

Perform the following steps:



**Step 1** Set [I2C\\_GLB\[i2c\\_enable\]](#) to 0 to disable the controller.

**Step 2** Set [I2C\\_INTR\\_EN](#) to 0x0 and [I2C\\_INTR\\_RAW](#) to 0x1fff to clear the interrupt and disable interrupt reporting, making preparation for switchover between operating modes.

----End

## 12.1.6 Register Summary

The I<sup>2</sup>C base addresses are as follows:

- PC0: 0x1106\_0000
- PC1: 0x1106\_1000
- PC2: 0x1106\_2000

[Table 12-7](#) describes the value range and meaning of the variables in the offset addresses for I<sup>2</sup>C registers.

**Table 12-7** Variables in the I<sup>2</sup>C register offset addresses

Variable	Value Range	Description
n	0–31	I <sup>2</sup> C timing command. A maximum of 32 timing commands are supported.

**Table 12-8** Summary of I<sup>2</sup>C registers

Offset Address	Register	Description	Page
0x0000	I2C_GLB	I <sup>2</sup> C global configuration register	<a href="#">12-16</a>
0x0004	I2C_HCNT	I <sup>2</sup> C high level duration register	<a href="#">12-17</a>
0x0008	I2C_LCNT	I <sup>2</sup> C low level duration register	<a href="#">12-17</a>
0x0010	I2C_DEV_ADDR	I <sup>2</sup> C component address register	<a href="#">12-18</a>
0x0014	I2C_DATA_BUF	I <sup>2</sup> C data buffer register	<a href="#">12-18</a>
0x0018	I2C_PATTERN_DATA1	I <sup>2</sup> C pattern data 1 register	<a href="#">12-19</a>
0x001C	I2C_PATTERN_DATA2	I <sup>2</sup> C pattern data 2 register	<a href="#">12-19</a>
0x0020	I2C_TX_FIFO	I <sup>2</sup> C TX FIFO data register	<a href="#">12-19</a>
0x0024	I2C_RX_FIFO	I <sup>2</sup> C RX FIFO data register	<a href="#">12-20</a>
0x0030 + nx4	I2C_TIMING_CMD	I <sup>2</sup> C timing command register	<a href="#">12-20</a>



Offset Address	Register	Description	Page
0x00B0	I2C_LOOP1	I <sup>2</sup> C loop number 1 register	<a href="#">12-22</a>
0x00B4	I2C_DST1	I <sup>2</sup> C jump destination 1 register	<a href="#">12-22</a>
0x00B8	I2C_LOOP2	I <sup>2</sup> C loop number 2 register	<a href="#">12-22</a>
0x00BC	I2C_DST2	I <sup>2</sup> C jump destination 2 register	<a href="#">12-23</a>
0x00C0	I2C_LOOP3	I <sup>2</sup> C loop number 3 register	<a href="#">12-23</a>
0x00C4	I2C_DST3	I <sup>2</sup> C jump destination 3 register	<a href="#">12-23</a>
0x00C8	I2C_TX_WATERMARK	I <sup>2</sup> C TX FIFO watermark register	<a href="#">12-24</a>
0x00CC	I2C_RX_WATERMARK	I <sup>2</sup> C RX FIFO watermark register	<a href="#">12-24</a>
0x00D0	I2C_CTRL1	I <sup>2</sup> C control register 1	<a href="#">12-24</a>
0x00D4	I2C_CTRL2	I <sup>2</sup> C control register 2	<a href="#">12-25</a>
0x00D8	I2C_FIFO_STAT	I <sup>2</sup> C FIFO status register	<a href="#">12-26</a>
0x00E0	I2C_INTR_RAW	I <sup>2</sup> C raw interrupt register	<a href="#">12-27</a>
0x00E4	I2C_INTR_EN	I <sup>2</sup> C interrupt enable register	<a href="#">12-28</a>
0x00E8	I2C_INTR_STAT	I <sup>2</sup> C interrupt status register	<a href="#">12-29</a>

## 12.1.7 Register Description

### I2C\_GLB

I2C\_GLB is an I<sup>2</sup>C global configuration register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:8]	RW	sda_hold_duration	SDA hold time duration Recommended value: <b>0xa</b> Note: It controls only the hold time duration of data sent from the host to the slave device.	0x0000
[7:1]	-	reserved	Reserved	0x00
[0]	RW	i2c_enable	I <sup>2</sup> C enable control	0x0



Bits	Access	Name	Description	Reset
			0: disabled. TX FIFO and RX FIFO are cleared. 1: enabled Note: If timing exceptions occur, disable and then enable the controller in case that residual data of the previous operation exists in the TX FIFO and RX FIFO.	

## I2C\_HCNT

I2C\_HCNT is an I<sup>2</sup>C high level duration register.

Offset Address: 0x0004 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	i2c_high_duration	SCL high level duration It is recommended that i2c_high_duration be 1/2 of the SCL cycle in standard mode and 0.36 times of the SCL cycle in fast mode. Standard mode: i2c_high_duration = $(F_{I2C\_CLK}/F_{SCL}) \times 0.5$ Fast mode: i2c_high_duration = $(F_{I2C\_CLK}/F_{SCL}) \times 0.36$ ( $F_{SCL}$ is the SCL bus frequency.) Assume that the reference clock is 50 MHz and works in fast mode ( $F_{SCL} = 400$ kHz): i2c_high_duration = $(50\text{ MHz}/400\text{ kHz}) \times 0.36 = 45$	0x0000

## I2C\_LCNT

I2C\_LCNT is an I<sup>2</sup>C low level duration register.

Offset Address: 0x0008 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	i2c_low_duration	SCL low level duration The reference clock is the bus clock SYSAPB.  It is recommended that i2c_low_duration be 1/2 of the SCL cycle in standard mode and 0.64 times of the SCL cycle in fast mode.  Standard mode: $i2c\_low\_duration = (F_{I2C\_CLK}/F_{SCL}) \times 0.5$ Fast mode: $i2c\_low\_duration = (F_{I2C\_CLK}/F_{SCL}) \times 0.64$ ( $F_{SCL}$ is the SCL bus frequency.)  Assume that the reference clock is 50 MHz and works in fast mode ( $F_{SCL} = 400$ kHz): $i2c\_low\_duration = (50 \text{ MHz}/400 \text{ kHz}) \times 0.64 = 80$	0x0000

## I2C\_DEV\_ADDR

I2C\_DEV\_ADDR is an I<sup>2</sup>C component address register.

Offset Address: 0x0010 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	dev_addr_byte4	Byte 4 of the component address	0x00
[23:16]	RW	dev_addr_byte3	Byte 3 of the component address	0x00
[15:8]	RW	dev_addr_byte2	Byte 2 of the component address	0x00
[7:0]	RW	dev_addr_byte1	Byte 1 of the component address	0x00

## I2C\_DATA\_BUF

I2C\_DATA\_BUF is an I<sup>2</sup>C data buffer register.

Offset Address: 0x0014 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:24]	RW	data_buf_byte4	Byte 4 of the data buffer	0x00
[23:16]	RW	data_buf_byte3	Byte 3 of the data buffer	0x00
[15:8]	RW	data_buf_byte2	Byte 2 of the data buffer	0x00
[7:0]	RW	data_buf_byte1	Byte 1 of the data buffer	0x00

## I2C\_PATTERN\_DATA1

I2C\_PATTERN\_DATA1 is an I<sup>2</sup>C pattern data 1 register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	pattern_data_byte4	Byte 4 of pattern data	0x00
[23:16]	RW	pattern_data_byte3	Byte 3 of pattern data	0x00
[15:8]	RW	pattern_data_byte2	Byte 2 of pattern data	0x00
[7:0]	RW	pattern_data_byte1	Byte 1 of pattern data	0x00

## I2C\_PATTERN\_DATA2

I2C\_PATTERN\_DATA2 is an I<sup>2</sup>C pattern data 2 register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	RW	pattern_data_byte8	Byte 8 of pattern data	0x00
[23:16]	RW	pattern_data_byte7	Byte 7 of pattern data	0x00
[15:8]	RW	pattern_data_byte6	Byte 6 of pattern data	0x00
[7:0]	RW	pattern_data_byte5	Byte 5 of pattern data	0x00

## I2C\_TX\_FIFO

I2C\_TX\_FIFO is an I<sup>2</sup>C TX FIFO data register.

Offset Address: 0x0020 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	WO	tx_fifo	TX FIFO entry The written data is data to be transmitted.	0x00

## I2C\_RX\_FIFO

I2C\_RX\_FIFO is an I<sup>2</sup>C RX FIFO data register.

Offset Address: 0x0024 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RO	rx_fifo	RX FIFO egress The read data is data received from the I <sup>2</sup> C bus.	0x00

## I2C\_TIMING\_CMD

I2C\_TIMING\_CMD is an I<sup>2</sup>C timing command register.

Offset Address: 0x0030 +  $n \times 4$  ( $n = 0-31$ ) Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x00000000
[4:0]	RW	timing_cmd	Timing command. $n = 0, 1, 2, \dots, 31$ . 0x00: EXIT (End. It is used to exit the logic.) 0x01: S (bus START) 0x02: WDA4 (Send dev_addr_byte4.) 0x03: WDA3 (Send dev_addr_byte3.) 0x04: WDA2 (Send dev_addr_byte2.) 0x05: WDA1 (Send dev_addr_byte1.) 0x06: WDB4 (Send data_buf_byte4.) 0x07: WDB3 (Send data_buf_byte3.) 0x08: WDB2 (Send data_buf_byte2.)	0x00



Bits	Access	Name	Description	Reset
			<p>0x09: WDB1 (Send data_buf_byte1.)</p> <p>0x0A: WPD8 (Send pattern_data_byte8.)</p> <p>0x0B: WPD7 (Send pattern_data_byte7.)</p> <p>0x0C: WPD6 (Send pattern_data_byte6.)</p> <p>0x0D: WPD5 (Send pattern_data_byte5.)</p> <p>0x0E: WPD4 (Send pattern_data_byte4.)</p> <p>0x0F: WPD3 (Send pattern_data_byte3.)</p> <p>0x10: WPD2 (Send pattern_data_byte2.)</p> <p>0x11: WPD1 (Send pattern_data_byte1.)</p> <p>0x12: RD (Receive 1 byte data.)</p> <p>0x13: RACK (Receive low-level acknowledgement.)</p> <p>0x14: RNACK (Receive high-level non-acknowledgement.)</p> <p>0x15: RNC (Receive acknowledgement regardless of the level.)</p> <p>0x16: SACK (Send low-level acknowledgement.)</p> <p>0x17: SNACK (Send high-level non-acknowledgement.)</p> <p>0x18: JMPN1 (Jump for limited times. The destination is specified by the DST1 register, and the number of jump times is specified by the LOOP1 register.)</p> <p>0x19: JMPN2 (Jump for limited times. The destination is specified by the DST2 register, and the number of jump times is specified by the LOOP2 register.)</p> <p>0x1A: JMPN3 (Jump for limited times. The destination is specified by the DST3 register, and the number of jump times is specified by the LOOP3 register.)</p> <p>0x1B: UNDEF (undefined)</p> <p>0x1C: UNDEF (undefined)</p> <p>0x1D: UDB1 (Update date from the TX FIFO to data_buf_byte1.)</p> <p>0x1E: SR (bus repeated START)</p> <p>0x1F: P (bus STOP)</p> <p><b>Note the following:</b></p>	



Bits	Access	Name	Description	Reset
			<ul style="list-style-type: none"><li>When the UDB1 command is executed, if the TX FIFO is empty, the controller waits until the TX FIFO has data. During waiting, the controller does not change the status of the I<sup>C</sup> bus.</li><li>When the RD command is executed, if the RX FIFO is full, the controller waits until the RX FIFO has spare space. During waiting, the controller does not change the status of the I<sup>C</sup> bus.</li></ul>	

## I2C\_LOOP1

I2C\_LOOP1 is an I<sup>C</sup> loop number 1 register.

Offset Address: 0x00B0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	loop_num1	Number of loops	0x0000_0000

## I2C\_DST1

I2C\_DST1 is an I<sup>C</sup> jump destination 1 register.

Offset Address: 0x00B4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	RW	reserved	Reserved	0x00000000
[4:0]	RW	dst_timing_cmd1	Jump destination timing command register	0x00

## I2C\_LOOP2

I2C\_LOOP2 is an I<sup>C</sup> loop number 2 register.

Offset Address: 0x00B8 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	loop_num2	Number of loops	0x0000_0000

## I2C\_DST2

I2C\_DST2 is an I<sup>2</sup>C jump destination 2 register.

Offset Address: 0x00BC Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	RW	reserved	Reserved	0x00000000
[4:0]	RW	dst_timing_cmd2	Jump destination timing command register	0x00

## I2C\_LOOP3

I2C\_LOOP3 is an I<sup>2</sup>C loop number 3 register.

Offset Address: 0x00C0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	loop_num3	Number of loops	0x0000_0000

## I2C\_DST3

I2C\_DST3 is an I<sup>2</sup>C jump destination 3 register.

Offset Address: 0x00C4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	RW	reserved	Reserved	0x00000000
[4:0]	RW	dst_timing_cmd3	Jump destination timing command register	0x00



## I2C\_TX\_WATERMARK

I2C\_TX\_WATERMARK is an I<sup>2</sup>C TX FIFO watermark register.

Offset Address: 0x00C8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x000000
[5:0]	RW	tx_watermark	TX FIFO watermark. When data in the TX FIFO is less than that specified by tx_watermark, the tx_lt_watermark_raw raw interrupt is set to 1.	0x00

## I2C\_RX\_WATERMARK

I2C\_RX\_WATERMARK is an I<sup>2</sup>C RX FIFO watermark register.

Offset Address: 0x00CC Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	RO	reserved	Reserved	0x000000
[5:0]	RW	rx_watermark	RX FIFO watermark. When data in the RX FIFO is more than that specified by rx_watermark, the rx_gt_watermark_raw raw interrupt is set to 1.	0x00

## I2C\_CTRL1

I2C\_CTRL1 is I<sup>2</sup>C control register 1.

Offset Address: 0x00D0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x000000
[9:8]	RW	dma_operation	DMA operation control 00: non-DMA mode 01: reserved 10: DMA write mode 11: DMA read mode	0x0
[7:1]	-	reserved	Reserved	0x00



Bits	Access	Name	Description	Reset
[0]	RW	start	Boot control. Writing 1 to this bit enables the state machine inside the controller to execute the timing command sequence. After execution is complete or an exception occurs, the logic clears this bit. 0: not boot 1: boot	0x0

## I2C\_CTRL2

I2C\_CTRL2 is I<sup>2</sup>C control register 2.

Offset Address: 0x00D4 Total Reset Value: 0x1111\_0011

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x0
[28]	RO	i2c_scl_oen	SDA output level of the internal I <sup>2</sup> C 0: low level 1: high level	0x1
[27:25]	-	reserved	Reserved	0x0
[24]	RO	i2c_sda_oen	SCL output level of the internal I <sup>2</sup> C 0: low level 1: high level	0x1
[23:21]	RO	reserved	Reserved	0x0
[20]	RO	i2c_scl_in	SCL level of the external I <sup>2</sup> C bus 0: low level 1: high level	0x1
[19:17]	-	reserved	Reserved	0x0
[16]	RO	i2c_sda_in	SDA level of the external I <sup>2</sup> C bus 0: low level 1: high level	0x1
[15:9]	-	reserved	Reserved	0x00
[8]	RW	gpio_mode	General-purpose input/output (GPIO) mode enabling for the SCL and SDA pins of the I <sup>2</sup> C (In this mode, the pin level is	0x0



Bits	Access	Name	Description	Reset
			determined by force_scl_oe_n and force_sda_oe_n.) 0: disabled 1: enabled	
[7:5]	-	reserved	Reserved	0x0
[4]	RW	force_scl_oen	SCL pin levelin GPIO mode 0: low level 1: high level	0x1
[3:1]	-	reserved	Reserved	0x0
[0]	RW	force_sda_oen	SDA pin levelin GPIO mode 0: low level 1: high level	0x1

## I2C\_FIFO\_STAT

I2C\_FIFO\_STAT is an I<sup>2</sup>C FIFO status register.

Offset Address: 0x00D8 Total Reset Value: 0x000A\_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x0000
[19]	RO	tx_fifo_not_full	TX FIFO full indicator 0: full 1: not full	0x1
[18]	RO	tx_fifo_not_empty	TX FIFO empty indicator 0: empty 1: not empty	0x0
[17]	RO	rx_fifo_not_full	RX FIFO full indicator 0: full 1: not full	0x1
[16]	RO	rx_fifo_not_empty	RX FIFO empty indicator 0: empty 1: not empty	0x0
[15]	-	reserved	Reserved	0x0



Bits	Access	Name	Description	Reset
[14: 8]	RO	tx_fifo_vld_num	Valid data number in the TX FIFO	0x00
[7]	-	reserved	Reserved	0x0
[6: 0]	RO	rx_fifo_vld_num	Valid data number in the RX FIFO	0x00

## I2C\_INTR\_RAW

I2C\_INTR\_RAW is an I<sup>2</sup>C raw interrupt register.

Offset Address: 0x00E0 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12]	RWC	all_cmd_done_raw	Interrupt indicating that the EXIT command in the timing command sequence is executed 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[11]	RWC	arb_lost_raw	Arbitration loss interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[10]	RWC	start_det_raw	Interrupt indicating that START is detected 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[9]	RWC	stop_det_raw	Interrupt indicating that STOP is detected 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8:5]	-	reserved	Reserved	0x0
[4]	RWC	tx_lt_watermark_rw	Interrupt indicating that the data amount in the TX FIFO is less than the watermark during data transmission 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	RO	reserved	Reserved	0x0
[2]	RWC	rx_gt_watermark_r	Interrupt indicating that the data amount	0x0



Bits	Access	Name	Description	Reset
		aw	in the RX FIFO is greater than the watermark during data reception 0: No interrupt is generated. 1: An interrupt is generated.	
[1]	-	reserved	Reserved	0x0
[0]	RWC	ack_bit_unmatch_rw	Interrupt indicating that the acknowledgement bit is not as expected 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## I2C\_INTR\_EN

I2C\_INTR\_EN is an I<sup>2</sup>C interrupt enable register.

Offset Address: 0x00E4 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12]	RW	all_cmd_done_en	all_cmd_done interrupt enable 0: disabled 1: enabled	0x0
[11]	RW	arb_lost_en	arb_lost interrupt enable 0: disabled 1: enabled	0x0
[10]	RW	start_det_en	start_det interrupt enable 0: disabled 1: enabled	0x0
[9]	RW	stop_det_en	STOP interrupt enable 0: disabled 1: enabled	0x0
[8:5]	-	reserved	Reserved	0x0
[4]	RW	tx_lt_watermark_en	tx_lt_watermark interrupt enable 0: disabled 1: enabled	0x0



Bits	Access	Name	Description	Reset
[3]	-	reserved	Reserved	0x0
[2]	RW	rx_gt_watermark_en	rx_gt_watermark interrupt enable 0: disabled 1: enabled	0x0
[1]	-	reserved	Reserved	0x0
[0]	RW	ack_bit_unmatch_en	ack_bit_unmatch interrupt enable 0: disabled 1: enabled	0x0

## I2C\_INTR\_STAT

I2C\_INTR\_STAT is an I<sup>2</sup>C interrupt status register.

Offset Address: 0x00E8 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12]	RO	all_cmd_done	Interrupt indicating that all timing commands are completed properly 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[11]	RO	arb_lost	Arbitration loss interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[10]	RO	start_det	Interrupt indicating that START is detected 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[9]	RO	stop_det	Interrupt indicating that STOP is detected 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8:5]	-	reserved	Reserved	0x0
[4]	RO	tx_lt_watermark	Interrupt indicating that the data amount in the TX FIFO is less than the watermark during data transmission	0x0



Bits	Access	Name	Description	Reset
			0: No interrupt is generated. 1: An interrupt is generated.	
[3]	-	reserved	Reserved	0x0
[2]	RO	rx_gt_watermark	Interrupt indicating that the data amount in the RX FIFO is greater than the watermark during data reception 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	-	reserved	Reserved	0x0
[0]	RO	ack_bit_unmatch	Interrupt indicating that the acknowledgement bit is not as expected 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## 12.2 UART

### 12.2.1 Overview

The universal asynchronous receiver transmitter (UART) is an asynchronous serial communication interface. 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 is mainly used to interconnect Hi3519DV500 with the UART of an external chip so that the two chips can communicate with each other.

The SoC provides three UART controllers:

- UART0: 2-wire UART for debugging
- UART1/2: 2/4-wire UART (for details, see the *Hi35xxVxxx\_PINOUT\_EN*)



## NOTICE

- ⚠ When the UART function is used, the RXD pin needs to be configured as pull-up enable before the UART is initialized. For details about the configuration registers, see the *Hi35xxVxxx\_PINOUT\_EN.xls*. (Hi35xxVxxx indicates an SoC version.)
- ⚠ For example, when UART0 of the system is used, bit[8] of the iocfg\_reg22 register needs to be set to 1.

### 12.2.2 Features

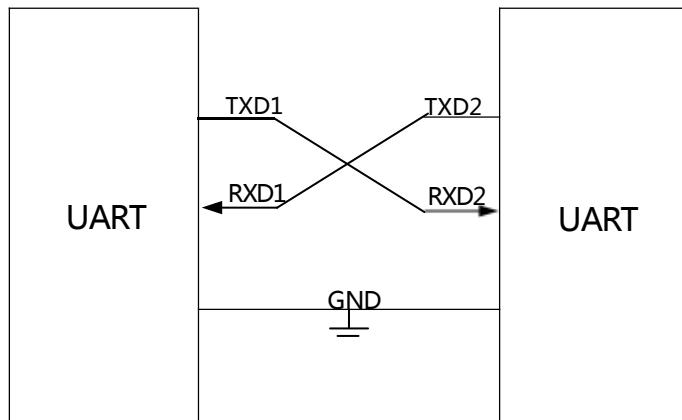
The UART unit has the following features:

- Supports 64x8-bit transmit first-in, first-out (FIFO) and 64x12-bit RX FIFO.
- Supports programmable widths for the data bit and stop bit. The width of the data bit can be set to 5 bits, 6 bits, 7 bits, or 8 bits and width of the stop bit can be set to 1 bit or 2 bits by programming.
- Supports parity check or no check.
- Supports the programmable transfer rate.
- Supports RX FIFO interrupts, TX FIFO interrupts, RX timeout interrupts, and error interrupts.
- Supports the query of the raw interrupt status and the masked interrupt status.
- Allows the UART or the TX/RX function of the UART to be disabled by programming to reduce power consumption.
- Allows the UART clock to be disabled to reduce power consumption.
- Supports DMA operations.

### 12.2.3 Function Description

#### Application Block Diagram

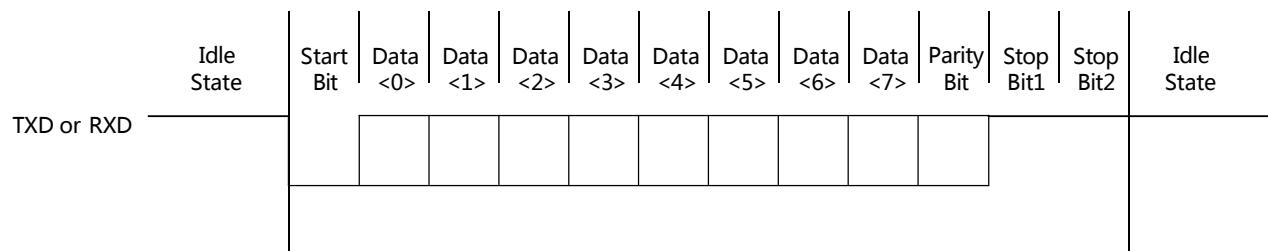
[Figure 12-7](#) shows the typical application of the UART.

**Figure 12-7** Typical application block diagram of the UART

The UART serves as an asynchronous bidirectional serial bus. Through the UARTs connected by two data lines, a simplified and effective data transfer mode is implemented.

## Function Principle

A frame transfer of the UART involves the start signal, data signal, parity bit, and stop signal, as shown in [Figure 12-8](#). The data frame is output from the TxD end of one UART and then is input to the RxD end of the other UART.

**Figure 12-8** Frame format of the UART

The definitions of the start signal, data signal, parity bit, and stop signal are as follows:

- 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 does not transmit data, the level must remain high.

- Data signal (data bit)

The data bit width can be set to 5 bits, 6 bits, 7 bits, or 8 bits according to the requirements in different applications.

- Parity bit



It is a 1-bit error correction signal. The UART parity bit can be an odd parity bit, even parity bit, or stick parity bit. The UART can enable and disable the parity bit. For details, see the description of the [UART\\_LCR\\_H](#) register.

- Stop signal (stop bit)

It is the stop bit of a data frame. The stop bit width can be set to 1 bit or 2 bits. The high level of TXD indicates the end of the data frame.

## 12.2.4 Operating Mode

### 12.2.4.1 Baud Rate Configuration

The operating baud rate of the UART can be set by configuring the registers

[UART\\_IBRD](#) and [UART\\_FBRD](#). The baud rate is calculated as follows:

Current baud rate = Frequency of the UART reference clock/(16 x Clock divider). The frequency of the UART reference clock is 100 MHz, 50 MHz, 24 MHz, or 3 MHz.

The clock frequency divider consists of the integral part and the fractional part that correspond to [UART\\_IBRD](#) and [UART\\_FBRD](#) respectively.

For example, assume that the frequency of the UART reference clock is 24 MHz. If [UART\\_IBRD](#) is set to 0x1E and [UART\\_FBRD](#) is set to 0x00, the current baud rate is calculated as follows:  $24/(16 \times 30) = 0.05$  Mbit/s

The typical UART baud rates are 9600 bit/s, 14,400 bit/s, 19,200 bit/s, 38,400 bit/s, 57,600 bit/s, 76,800 bit/s, 115,200 bit/s, 230,400 bit/s, and 460,800 bit/s.

The following examples show how to calculate the clock divider and how to configure the clock divider register:

If the required baud rate is 230,400 bit/s and the frequency of the UART reference clock is 24 MHz, the clock divider is calculated as follows:  $(24 \times 10^6)/(16 \times 230,400) = 6.5104$ . The integral part IBRD is 6 and the fractional part FBRD is 0.5104.

To calculate the value of the 6-bit [UART\\_FBRD](#) register, do as follows: calculate the value of  $m$  by using the following formula:  $m = \text{integer} (FBRD \times 2^n + 0.5) = (0.5104 \times 2^6 + 0.5) = 33$  ( $n$  = the width of [UART\\_FBRD](#)). Then set [UART\\_IBRD](#) to 0x0006 and set [UART\\_FBRD](#) to 0x21.

If the fractional part of the frequency divider is set to 33, the actual divisor of the baud rate is  $6 + 33/2^6 = 6.5156$ , the baud rate is  $(24 \times 16)/(16 \times 605156) = 230,216.7107$ , and the error rate is  $(230,216.7107 - 230,400)/230,400 \times 100 = -0.07956\%$ .

The maximum error rate is  $1/2^6 \times 100 = 1.56\%$  when the 6-bit [UART\\_FBRD](#) is used. If the value of  $m$  is 1, the total error rate is greater than 64 clock cycles.

### 12.2.4.2 Soft Reset



For details about reset registers, see section 3.2.5 "Register Description."



After reset, the configuration registers are restored to default values. Therefore, these registers must be initialized again.

### 12.2.4.3 Data Transfer in Interrupt or Query Mode

#### Initialization

The initialization is implemented as follows:

- Step 1** Write 0 to [UART\\_CR](#)bit[0] to disable the UART.
- Step 2** Write to [UART\\_IBRD](#)and [UART\\_FBRD](#)to configure the transfer rate.
- Step 3** Configure [UART\\_CR](#)and [UART\\_LCR\\_H](#)to set the UART operating mode.
- Step 4** Configure [UART\\_IFLS](#)to set the thresholds of TX and RX FIFOs.
- Step 5** If the driver runs in interrupt mode, set [UART\\_IMSC](#)to enable the corresponding interrupt; if the driver runs in query mode, disable the generation of corresponding interrupts.
- Step 6** Write 1 to [UART\\_CR](#)bit[0] to enable the UART.

----End

#### Data Transmission

To transmit data, perform the following steps:

- Step 1** Write the data to be transmitted to [UART\\_DR](#)and start data transmission.
- Step 2** In query mode, check the TX\_FIFO status by reading [UART\\_FR](#)bit[5] during the continuous data transmission. According to the TX\_FIFO status, determine whether to transmit data to TX\_FIFO. In interrupt mode, check the TX\_FIFO status by reading the corresponding interrupt status bits and then determine whether to transmit data to TX\_FIFO.
- Step 3** Check whether the UART transmits all data by reading [UART\\_FR](#)bit[7]. If [UART\\_FR](#) bit[7] is 1, the UART transmits all data.

----End

#### Data Reception

To receive data, perform the following steps:

- In query mode, detect the RX\_FIFO status by reading [UART\\_FR](#)[rxe] during data reception and then determine whether to read data from the RX\_FIFO according to the RX\_FIFO status.
- In interrupt mode, determine whether to read data from RX\_FIFO according to corresponding interrupt status bits.



## 12.2.4.4 Data Transfer in DMA Mode

### Initialization

To initialize the UART, perform the following steps:

- Step 1** Write 0 to [UART\\_CR\[uarten\]](#) to disable the UART.
- Step 2** Write values to [UART\\_IBRD](#) and [UART\\_FBRD](#) to configure the data transfer rate.
- Step 3** Configure [UART\\_CR](#) and [UART\\_LCR\\_H](#) to set the UART operating mode.
- Step 4** Configure [UART\\_IFLS](#) to set the TX and RX FIFO thresholds.
- Step 5** If the driver runs in interrupt mode, set [UART\\_IMSC](#) to enable corresponding interrupts. If the driver runs in query mode, disable the generation of corresponding interrupts.
- Step 6** Write 1 to [UART\\_CR\[uarten\]](#) to enable the UART.

----End

### Data Transfer

To transmit data, perform the following steps (the following is an example using the DMA to transmit data):

- Step 1** Configure the DMA data channel, including the transfer source address, transfer destination address, number of data items to be transferred, and transfer type. For details, see the description in xx "DMAC."
- Step 2** Set [UART\\_DMACR](#) to 0x2 to enable the DMA transfer function of the UART.
- Step 3** Check whether the data is transferred completely based on the interrupt report status of the DMA. If all data is transferred, disable the DMA transfer function of the UART.

----End

### Data Reception

To receive data, perform the following steps (the following is an example using the DMA to receive data):

- Step 1** Configure the DMA data channel, including data transfer source and destination addresses, data receive area address, number of data items to be transferred, and transfer type.
- Step 2** Set [UART\\_DMACR](#) to 0x1 to enable the DMA receive function of the UART.
- Step 3** Check whether the data is received completely by querying the DMA status. If all data is received, disable the DMA receive function of the UART.

----End



## 12.2.5 Register Summary

The following lists the address space:

- The base address of UART0 registers is 0x1104\_0000.
- The base address of UART1 registers is 0x1104\_1000.
- The base address of UART2 registers is 0x1104\_2000.

Table 12-9 describes the UART registers.

**Table 12-9** Summary of UART registers

Offset Address	Register	Description	Page
0x000	UART_DR	Data register	<a href="#">12-36</a>
0x004	UART_RSR	Receive status register or error clear register	<a href="#">12-37</a>
0x018	UART_FR	Flag register	<a href="#">12-38</a>
0x024	UART_IBRD	Integral baud rate register	<a href="#">12-40</a>
0x028	UART_FBRD	Fractional baud rate register	<a href="#">12-40</a>
0x02C	UART_LCR_H	Line control register	<a href="#">12-40</a>
0x030	UART_CR	Control register	<a href="#">12-42</a>
0x034	UART_IFLS	Interrupt FIFO threshold select register	<a href="#">12-44</a>
0x038	UART_IMSC	Interrupt mask register	<a href="#">12-45</a>
0x03C	UART_RIS	Raw interrupt status register	<a href="#">12-46</a>
0x040	UART_MIS	Masked interrupt status register	<a href="#">12-47</a>
0x044	UART_ICR	Interrupt clear register	<a href="#">12-48</a>
0x048	UART_DMACR	DMA control register	<a href="#">12-48</a>

## 12.2.6 Register Description

### UART\_DR

UART\_DR is a UART data register that stores the received data and the data to be transmitted. The receive status can be queried by reading this register.

Offset Address: 0x000 Total Reset Value: 0x0000



Bits	Access	Name	Description	Reset
[15:12]	-	reserved	Reserved	0x0
[11]	RO	oe	Overflow error 0: No overflow error occurs. 1: An overflow error occurs. That is, a data segment is received when the RX FIFO is full.	0x0
[10]	RO	be	Break error 0: No break error occurs. 1: A break error occurs. That is, the time of RX data input signal keeping low is longer than a full word transfer. A full word consists of a start bit, a data bit, a parity bit, and a stop bit.	0x0
[9]	RO	pe	Parity error 0: No parity error occurs. 1: A parity error occurs.	0x0
[8]	RO	fe	Frame error 0: No frame error occurs. 1: A frame error (namely, stop bit error) occurs.	0x0
[7:0]	RW	data	Data received and to be transmitted	0x00

## UART\_RSR

UART\_RSR is a receive status register or error clear register.

- It acts as the receive status register when being read.
- It acts as the error clear register when being written.

You can query the receive status by reading [UART\\_DR](#). The status information about the break, frame, and parity read from [UART\\_DR](#) has priority over that read from [UART\\_RSR](#). That is, the status read from [UART\\_DR](#) changes faster than that read from [UART\\_RSR](#).

UART\_RSR is reset when any value is written to it.

Offset Address: 0x004 Total Reset Value: 0x00



Bits	Access	Name	Description	Reset
[7:4]	-	reserved	Reserved	0x0
[3]	RW	oe	Overflow error 0: No overflow error occurs. 1: An overflow error occurs. When the FIFO is full, the next data segment cannot be written to the FIFO and an overflow occurs in the shift register. Therefore, the contents in the FIFO are valid. In this case, the CPU must read the data immediately to spare the FIFO.	0x0
[2]	RW	be	Break error 0: No break error occurs. 1: A break error occurs. A break error occurs when the time of the RX data signal keeping low is longer than a full word transfer. A full word consists of a start bit, a data bit, a parity bit, and a stop bit.	0x0
[1]	RW	pe	Parity error 0: No parity error occurs. 1: A parity error occurs when the received data is checked. In FIFO mode, the error is associated with the data at the top of the FIFO.	0x0
[0]	RW	fe	Frame error 0: No frame error occurs. 1: The stop bit of the received data is incorrect. The valid stop bit is 1.	0x0

## UART\_FR

UART\_FR is a UART flag register.

Offset Address: 0x018 Total Reset Value: 0x0096

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x00



Bits	Access	Name	Description	Reset
[7]	RO	txfe	<p>The definition of the bit is determined by the status of <a href="#">UART_LCR_H</a> [fen].</p> <ul style="list-style-type: none"><li>• If <a href="#">UART_LCR_H</a> [fen] is 0, this bit is set to 1 when the TX holding register is empty.</li><li>• If <a href="#">UART_LCR_H</a> [fen] is 1, this bit is set to 1 when the TX FIFO is empty.</li></ul>	0x1
[6]	RO	rxff	<p>The definition of the bit is determined by the status of <a href="#">UART_LCR_H</a> [fen].</p> <ul style="list-style-type: none"><li>• If <a href="#">UART_LCR_H</a> [fen] is 0, this bit is set to 1 when the RX holding register is full.</li><li>• If <a href="#">UART_LCR_H</a> [fen] is 1, this bit is set to 1 when the RX FIFO is full.</li></ul>	0x0
[5]	RO	txff	<p>The definition of the bit is determined by the status of <a href="#">UART_LCR_H</a> [fen].</p> <ul style="list-style-type: none"><li>• If <a href="#">UART_LCR_H</a> [fen] is 0, this bit is set to 1 when the TX holding register is full.</li><li>• If <a href="#">UART_LCR_H</a> [fen] is 1, this bit is set to 1 when the TX FIFO is full.</li></ul>	0x0
[4]	RO	rxfe	<p>The definition of the bit is determined by the status of <a href="#">UART_LCR_H</a> [fen].</p> <ul style="list-style-type: none"><li>• If <a href="#">UART_LCR_H</a> [fen] is 0, this bit is set to 1 when the RX holding register is empty.</li><li>• If <a href="#">UART_LCR_H</a> [fen] is 1, this bit is set to 1 when the RX FIFO is empty.</li></ul>	0x1
[3]	RO	busy	<p>UART busy/idle status</p> <p>0: The UART is idle or data transmission is complete.</p> <p>1: The UART is busy in transmitting data. If the bit is set to 1, the status is kept until the entire byte (including all stop bits) is transmitted from the shift register.</p> <p>Regardless of whether the UART is enabled, this bit is set to 1 when the TX FIFO is not empty.</p>	0x0



Bits	Access	Name	Description	Reset
[2:0]	-	reserved	Reserved	0x6

## UART\_IBRD

UART\_IBRD is an integral baud rate register.

Offset Address: 0x024 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	baud divint	Clock frequency divider corresponding to the integral part of the baud rate. All bits are cleared after reset.	0x0000

## UART\_FBRD

UART\_FBRD is a fractional baud rate register.

### NOTICE

- ⚠ The values of UART\_IBRD and UART\_FBRD can be updated only after the current data is transmitted and received completely.
- ⚠ The minimum clock frequency divider is 1 and the maximum divider is 65,535 ( $2^{16} - 1$ ). That is, **UART\_IBRD** cannot be 0 and **UART\_FBRD** is ignored if **UART\_IBRD** is 0. Similarly, if **UART\_IBRD** is equal to 65,535 (0xFFFF), **UART\_IBRD** must be 0. If **UART\_FBRD** is greater than 0, the data fails to be transmitted or received.

Offset Address: 0x028 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:6]	-	reserved	Reserved	0x0
[5:0]	RW	band divfrac	Clock frequency divider corresponding to the fractional part of the baud rate. All bits are cleared after reset.	0x00



## UART\_LCR\_H

UART\_LCR\_H is a line control register. The registers [UART\\_LCR\\_H](#), [UART\\_IBRD](#), and [UART\\_FBRD](#) are combined to form a 30-bit register. If [UART\\_IBRD](#) and [UART\\_FBRD](#) are updated, [UART\\_LCR\\_H](#) must be updated at the same time.

Offset Address: 0x02C Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x00
[7]	RW	sps	<p>Stick parity select</p> <ul style="list-style-type: none"><li>When bit 1, bit 2, and bit 7 of this register are set to 1, the parity bit is 0 during transmission and detection.</li><li>When bit 1 and bit 7 of this register are set to 1 and bit 2 is set to 0, the parity bit is 1 during transmission and detection.</li><li>When the value is 0, the stick parity bit is disabled.</li></ul> <p>The parity check is disabled according to the current setting, and this bit is related only to stick parity.</p>	0x0
[6:5]	RW	wlen	<p>Count of bits in a transmitted or received frame</p> <p>00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits</p>	0x0
[4]	RW	fen	<p>TX/RX FIFO enable</p> <p>0: disabled 1: enabled</p>	0x0
[3]	RW	stp2	<p>2-bit stop bit at the end of a transmitted frame</p> <p>0: There is a 1-bit stop bit at the end of the transmitted frame. 1: There is a 2-bit stop bit at the end of the transmitted frame.</p> <p>The RX logic does not check for the 2-bit stop bit during data reception.</p>	0x0
[2]	RW	eps	Parity select during data transmission	0x0



Bits	Access	Name	Description	Reset
			and reception 0: The odd parity is generated or checked during data transmission and reception. 1: The even parity is generated or checked during data transmission and reception. When <a href="#">UART_LCR_H[pen]</a> is 0, this bit becomes invalid.	
[1]	RW	pen	Parity enable 0: The parity is disabled. 1: The parity is generated on the TX side and checked on the RX side.	0x0
[0]	RW	brk	Break transmit 0: invalid 1: After the current data transmission is complete, UTXD outputs low level continuously.   <b>NOTE</b> This bit must retain 1 during the period of at least two full frames to ensure the break command is executed properly. In general, the bit must be set to 0.	0x0

## UART\_CR

UART\_CR is a UART control register.

To configure UART\_CR, perform the following steps:

- Step 1** Write 0 to [UART\\_CR\[uarten\]](#) to disable the UART.
- Step 2** Wait until the current data transmission or reception is complete.
- Step 3** Clear [UART\\_LCR\\_H\[fen\]](#).
- Step 4** Configure [UART\\_CR](#).
- Step 5** Write 1 to [UART\\_CR\[uarten\]](#) to enable the UART.

----End

Offset Address: 0x030    Total Reset Value: 0x0300



Bits	Access	Name	Description	Reset
[15]	RW	ctsen	CTS hardware flow control enable 0: disabled 1: enabled. Data is transmitted only when the nUARTCTS signal is valid.	0x0
[14]	RW	rtsen	RTS hardware flow control enable 0: disabled 1: enabled. The data reception request is raised only when the RX FIFO has available space.	0x0
[13:12]	-	reserved	Reserved	0x0
[11]	RW	rts	Request transmit This bit is the inversion of the status output signal nUARTRTS of the UART modem. 0: The output signal retains. 1: When this bit is set to 1, the output signal is 0.	0x0
[10]	RW	dtr	Data transmit ready This bit is the inversion of the status output signal nUARTDTR of the UART modem. 0: The output signal retains. 1: When this bit is set to 1, the output signal is 0.	0x0
[9]	RW	rxen	UART receive enable 0: disabled 1: enabled If the UART is disabled during data reception, the current data reception is stopped abnormally.	0x1
[8]	RW	txen	UART transmit enable 0: disabled 1: enabled If the UART is disabled during data transmission, the current data transmission is stopped abnormally.	0x1
[7]	RW	lbe	Loopback enable	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: UARTRXD is looped back to UARTTXD.	
[6:1]	-	reserved	Reserved	0x00
[0]	RW	uarten	UART enable 0: disabled 1: enabled If the UART is disabled during data reception and transmission, the data transfer is stopped abnormally.	0x0

## UART\_IFLS

UART\_IFLS is an interrupt FIFO threshold select register. It is used to set a threshold for triggering a FIFO interrupt (UART\_TXINTR or UART\_RXINTR).

Offset Address: 0x034 Total Reset Value: 0x0012

Bits	Access	Name	Description	Reset
[15:6]	-	reserved	Reserved	0x000
[5:3]	RW	rxiflssel	RX interrupt FIFO threshold select. An RX interrupt is triggered when any of the following conditions is met: 000: RX FIFO $\geq$ 1/8 full 001: RX FIFO $\geq$ 1/4 full 010: RX FIFO $\geq$ 1/2 full 011: RX FIFO $\geq$ 3/4 full 100: RX FIFO $\geq$ 7/8 full 101: RX FIFO $\geq$ 1/16 full 110: RX FIFO $\geq$ 1/32 full 111: reserved	0x2
[2:0]	RW	txiflssel	TX interrupt FIFO threshold select. A TX interrupt is triggered when any of the following conditions is met: 000: TX FIFO $\leq$ 1/8 full 001: TX FIFO $\leq$ 1/4 full 010: TX FIFO $\leq$ 1/2 full 011: TX FIFO $\leq$ 3/4 full	0x2



Bits	Access	Name	Description	Reset
			100: TX FIFO $\leq$ 7/8 full 101: TX FIFO $\leq$ 15/16 full 110: TX FIFO $\leq$ 31/32 full 111: reserved	

## UART\_IMSC

UART\_IMSC is an interrupt mask register. It is used to mask interrupts.

Offset Address: 0x038 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	RW	oeim	Mask status of the overflow error interrupt 0: masked 1: not masked	0x0
[9]	RW	beim	Mask status of the break error interrupt 0: masked 1: not masked	0x0
[8]	RW	peim	Mask status of the parity interrupt 0: masked 1: not masked	0x0
[7]	RW	feim	Mask status of the frame error interrupt 0: masked 1: not masked	0x0
[6]	RW	rtim	Mask status of the RX timeout interrupt 0: masked 1: not masked	0x0
[5]	RW	txim	Mask status of the TX interrupt 0: masked 1: not masked	0x0
[4]	RW	rxim	Mask status of the RX interrupt 0: masked 1: not masked	0x0



Bits	Access	Name	Description	Reset
[3:0]	-	reserved	Reserved	0x0

## UART\_RIS

UART\_RIS is a raw interrupt status register. The contents of this register are not affected by the UART\_IMSC register.

Offset Address: 0x03C Total Reset Value: 0x0002

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	RO	oeris	Status of the raw overflow error interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[9]	RO	beris	Status of the raw break error interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	peris	Status of the raw parity interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7]	RO	feris	Status of the raw error interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	RO	rtris	Status of the raw RX timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[5]	RO	txris	Status of the raw TX interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	rxris	Status of the raw RX interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3:0]	-	reserved	Reserved	0x2



## UART\_MIS

UART\_MIS is a masked interrupt status register. The contents of this register are the results obtained after the raw interrupt status is ANDed with the interrupt mask status.

Offset Address: 0x040 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	RO	oemis	Status of the masked overflow error interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[9]	RO	bemis	Status of the masked break error interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	pemis	Status of the masked parity interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7]	RO	femis	Status of the masked error interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	RO	rtmis	Status of the masked RX timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[5]	RO	txmis	Status of the masked TX interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	rxmis	Status of the masked RX interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3:0]	-	reserved	Reserved	0x0



## UART\_ICR

UART\_ICR is an interrupt clear register. When 1 is written to it, the corresponding interrupt is cleared. Writing 0 has no effect.

Offset Address: 0x044 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	WO	oeic	Overflow error interrupt clear 0: invalid 1: cleared	0x0
[9]	WO	beic	Break error interrupt clear 0: invalid 1: cleared	0x0
[8]	WO	peic	Parity interrupt clear 0: invalid 1: cleared	0x0
[7]	WO	feic	Error interrupt clear 0: invalid 1: cleared	0x0
[6]	WO	rtic	RX timeout interrupt clear 0: invalid 1: cleared	0x0
[5]	WO	txic	TX interrupt clear 0: invalid 1: cleared	0x0
[4]	WO	rxic	RX interrupt clear 0: invalid 1: cleared	0x0
[3:0]	-	reserved	Reserved	0x0

## UART\_DMACR

UART\_DMACR is a DMA control register. It is used to enable or disable the DMAs of the TX and RX FIFOs.

Offset Address: 0x048 Total Reset Value: 0x0000



Bits	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x0000
[3]	RW	rxlastsreq_en	REQ enable of the last data supported by the UART RX DMA. After the function is enabled, the LASTREQ message is sent to the DMA controller when the last data is transferred. 0: disabled 1: enabled	0x0
[2]	RW	dmaonerr	DMA enable for the RX channel when the UART error interrupt (UARTEINTR) occurs. 0: When UARTEINTR is valid, the DMA output request (UARTRXDMASREQ or UARRTXDMABREQ) of the RX channel is valid. 1: When UARTEINTR is valid, the DMA output request (UARTRXDMASREQ or UARRTXDMABREQ) of the RX channel is invalid.	0x0
[1]	RW	txdmae	TX FIFO DMA enable 0: disabled 1: enable	0x0
[0]	RW	rxdmae	RX FIFO DMA enable 0: disabled 1: enable	0x0

## 12.3 eMMC/SDIO/SD Card Controller

### 12.3.1 Overview

The eMMC/SDIO/SD card controller (MMC for short) controls the read/write operations on the eMMC and SD card, and supports various extended devices such as Bluetooth and Wi-Fi devices based on the SDIO protocol. The SoC provides two MMCs: SDIO0 and SDIO1.

### 12.3.2 Features

The MMC has the following features:



#### Hi3516CV610 SDIO0:

- Can be connected to the eMMC, SD card, and SDIO Wi-Fi module.
- Data transmission in non-DMA, SDMA, and ADMA2 modes
- CRC generation and check for data and commands
- 1-bit, 4-bit, and 8-bit data widths, based on the connected component
- Read and write operations on data blocks with the size ranging from 1 byte to 65535 bytes.
- eMMC in DS, HS, HS200, and HS400 modes, and the boot function
- SD supports the default speed (DS) and high speed (HS) modes.
- SDIO supports the default speed (DS) and high speed (HS) modes.

#### NOTE

Hi3516CV610-10B/20B/20S/20G SDIO0 does not support the 8-bit data width, and the eMMC does not support the HS400 mode.

#### Hi3516CV610 SDIO1:

- Can be connected to the SD card and SDIO Wi-Fi module.
- Data transmission in non-DMA, SDMA, and ADMA2 modes
- CRC generation and check for commands and data.
- 1-bit and 4-bit data widths, based on the connected component
- Read and write operations on data blocks with the size ranging from 1 byte to 65535 bytes
- SD supports the default speed (DS) and high speed (HS) modes.
- SDIO supports the default speed (DS) and high speed (HS) modes.

### 12.3.3 Function Description

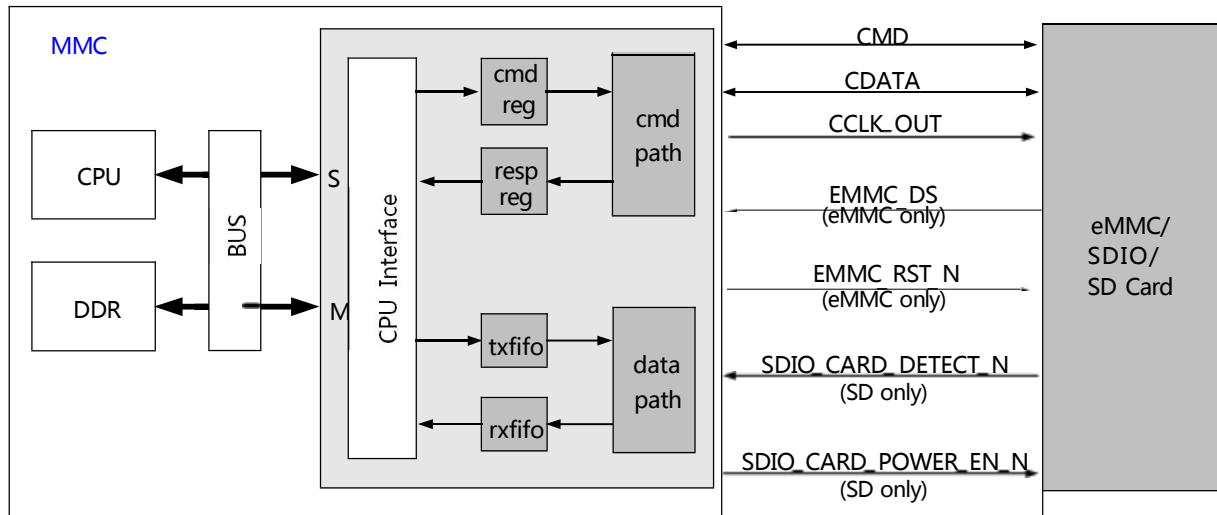
#### Functional Block Diagram

For details about the function signals and pins corresponding to the SDIO0 and SDIO1 controllers, see *Hi35xxVxxx\_PINOUT\_EN* (*Hi35xxVxxx* indicates an SoC version).

eMMC/SDIO/SD supports the devices that comply with the protocol:

- MultiMediaCard (eMMC-version 5.0)
- Secure Digital I/O (SDIO version 2.0).
- Secure Digital Memory (SD mem-version 2.0)

[Figure 12-9](#) shows the functional block diagram of the MMC controller.

**Figure 12-9** Functional block diagram of the MMC controller

### NOTE

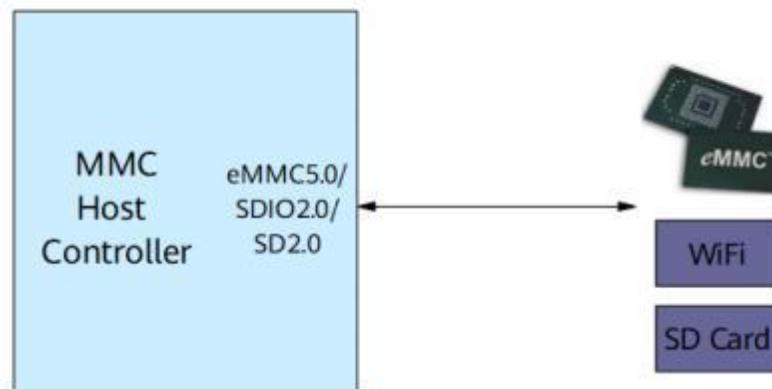
S indicates slave interface and M indicates master interface.

The MMC controller connects to the system through the internal bus, and consists of the following parts:

- Command path
  - Transmits commands and receives responses.
- Data Channel
  - Reads and writes data by working with the command path.
- Interface clock control unit
  - Enables and disables the clock, and adjusts the interface clock phase.

### Typical Application

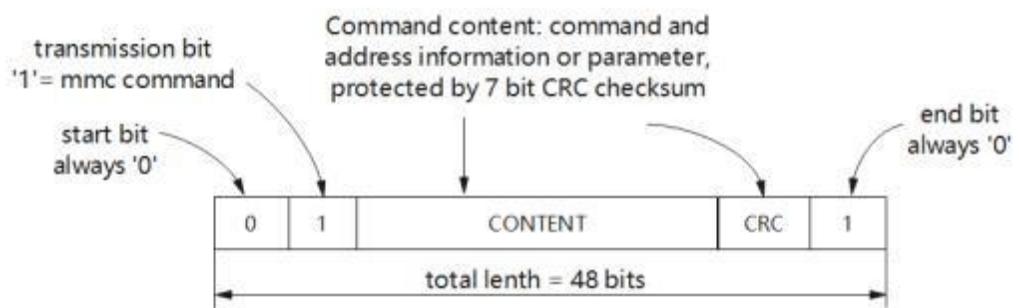
[Figure 12-10](#) shows the typical application of the MMC controller.

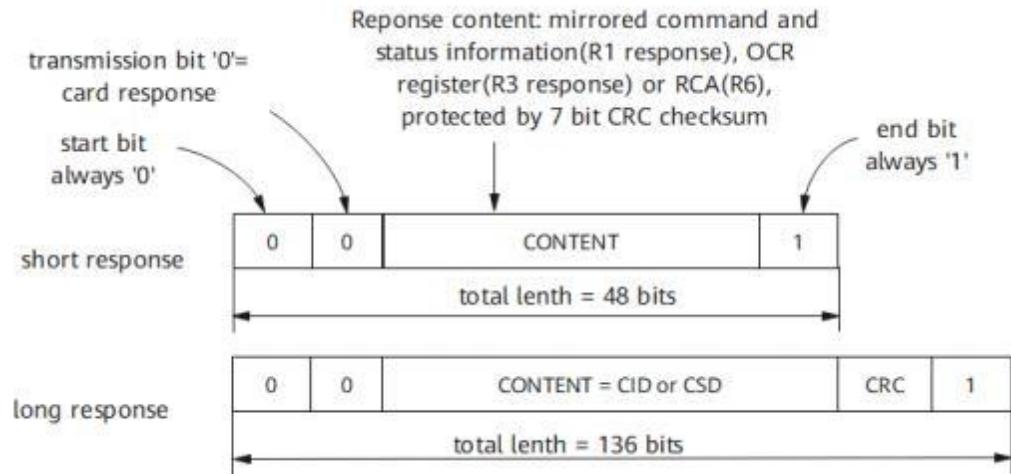
**Figure 12-10** Typical application of the MMC controller

## Commands and Responses

All interactions between the MMC controller and the card, including initializing the card, reading/writing to registers, querying the status, and transferring data, are implemented by using commands.

An MMC command is a segment of 48-bit serial data consisting of a start bit, a transmission bit, a command sequence, a CRC bit, and an end bit. After receiving a command, the card returns a 48-bit or a 136-bit response based on the command type.

**Figure 12-11** Format of an MMC command

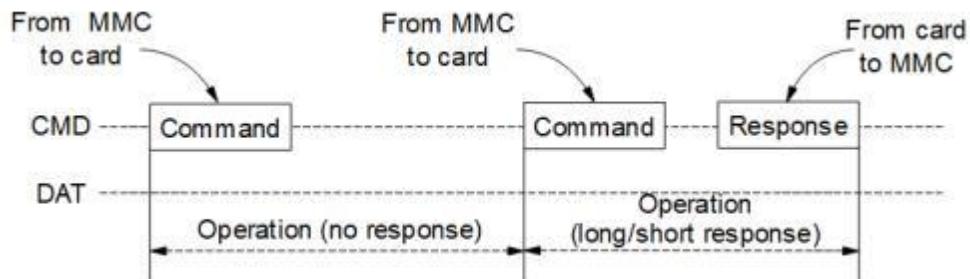
**Figure 12-12** Format of the response to an MMC command

Commands are classified into the following two types based on whether data is transferred:

- Non-data transfer command  
The MMC controller transmits/receives commands to/from the card through the command signal line.
- Data transfer command  
Besides the interaction on the command line, data is also transferred through the data lines DAT0 to DAT3 (eMMC DAT0 to DAT7).

### 1. Non-data transfer command

[Figure 12-13](#) shows the operations between the MMC controller and the card by running a non-data transfer command.

**Figure 12-13** Operations by running an MMC non-data transfer command

### 2. Data transfer command

The MMC supports the following data transfer commands:



- Stream data read/write command  
Only the MMC card supports the stream data read/write command. In this case, only the data line DAT0 is used for data transfer, and no CRC check is performed.
- Single-block read/write command  
After this command is executed, one single data block is read and written each time. No stop command is required for stopping each data transfer.
- Multi-block read/write command
  - Predefined block count mode  
Before the multi-block read/write command is executed, the block count command is transmitted to specify the number of data blocks to be transferred.
  - Open-ended mode  
After a read/write command is transmitted, a stop command is required for stopping data transfer at the end of data transfer.

The multi-block read/write command for the SDIO card is different from the preceding two modes. To be specific, the command parameter contains the number of data blocks to be transferred when the read/write command is transmitted.

Responses are classified into the following three types:

- Command without response  
For example, the card reset command.
- Short response command  
For example, the data transfer command and card status query command.
- Long response command  
This type of commands is used to read only the information about the card identification (CID) and card specific data (CSD) registers of a card.

## Data Transfer

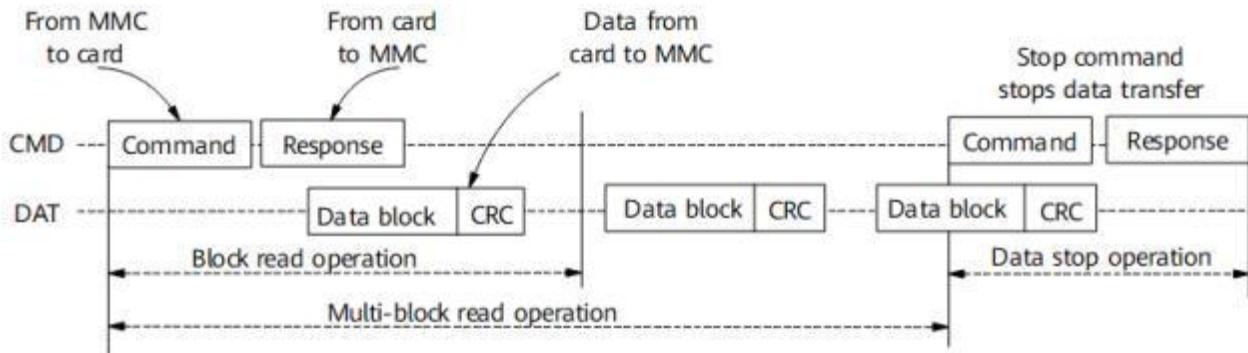
The single-block read/write command and the multi-block read/write command are widely used during data transfer. The data block during the data transfer of the eMMC is 512 bytes; the block size during the data transfer of the SDIO card can be customized

### NOTE

During the data transfer by using the block read/write command, the total data amount to be transferred must be an integral multiple of the block size.

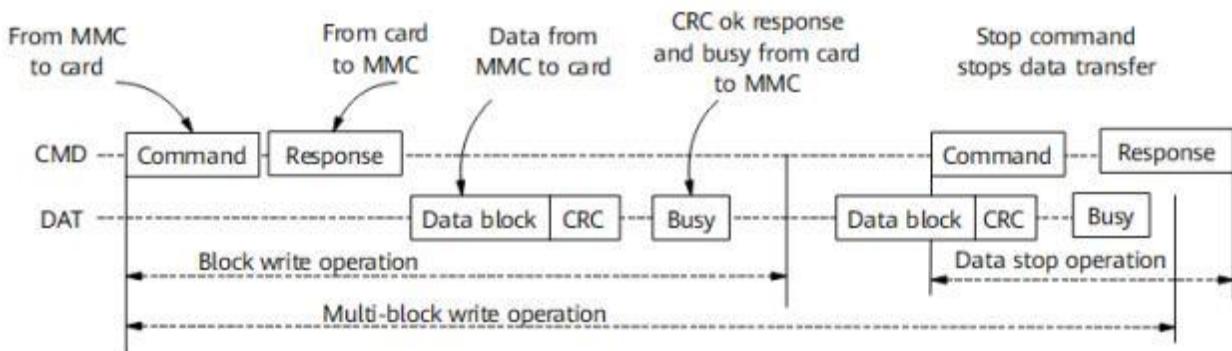
All data transfer commands are short response commands with data transferred through data lines. [Figure 12-14](#) and [Figure 12-15](#) show the relationships among the commands, responses, and timings of data lines.

#### 1. Single-block and multi-block read operations

**Figure 12-14 Single-block and multi-block read operations**

The MMC controller transmits a single-block or multi-block command to a card. When a response is being received, data is received by blocks. Each data block contains a CRC check bit for ensuring the integrity of the transferred data. In a single-block read operation, the data transfer is completed after the MMC controller receives a data block. In a multi-block read operation, the MMC controller needs to transmit a stop command to end the data transfer after receiving multiple data blocks only in open-ended mode.

## 2. Single-block and multi-block read operations

**Figure 12-15 Single-block and multi-block write operations**

The MMC controller transmits a single-block or multi-block command to a card. After receiving a response, the MMC controller starts to transmit data to the card by blocks. Each data block contains a CRC bit. The card performs CRC on each data block and transfers the CRC status to the MMC controller. This ensures that data is transferred properly.

In a single-block write operation, the data transfer is completed after the MMC controller transmits a data block. In a multi-block write operation, the MMC controller needs to transmit a stop command to end the data transfer after transmitting multiple data blocks only in open-ended mode. After a write operation, the card may be busy in programming the flash memory. The MMC

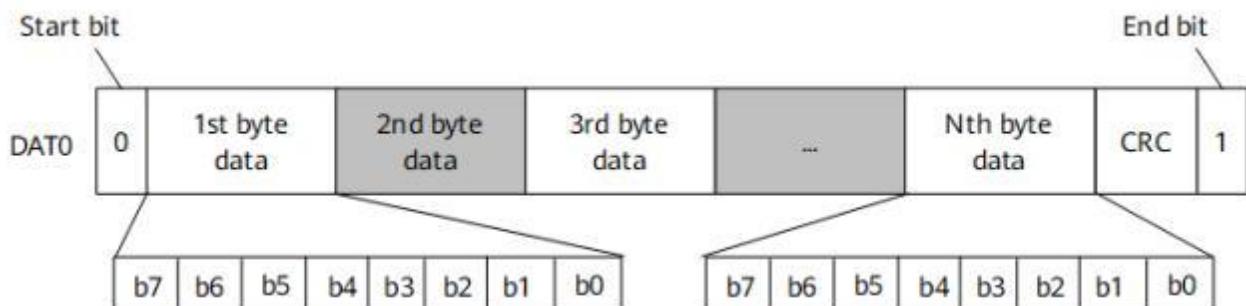
controller can perform the next operation on the card only after it confirms that the card is not busy by querying the status of the signal line DAT0.

### 3. Data transfer format

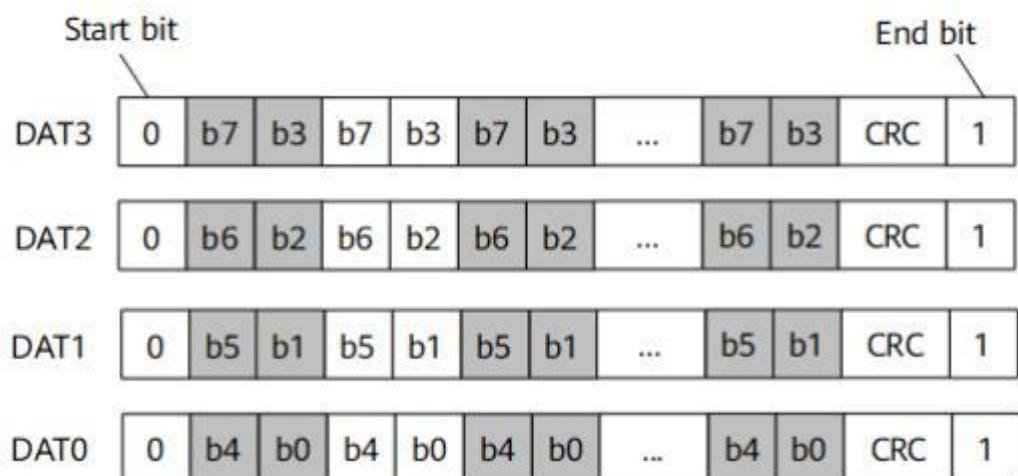
During the block read/write operations, the one or four data lines can be used to transfer data between the MMC controller and the card. Before a data transfer command is transmitted, the data transfer widths of the MMC controller and card must be the same, that is, 1 bit, 4 bits, or 8 bits (eMMC only). You can set the data bit width of the MMC controller by configuring **HOST\_CTRL1\_R** and set the data bit width of the card by transmitting the corresponding command.

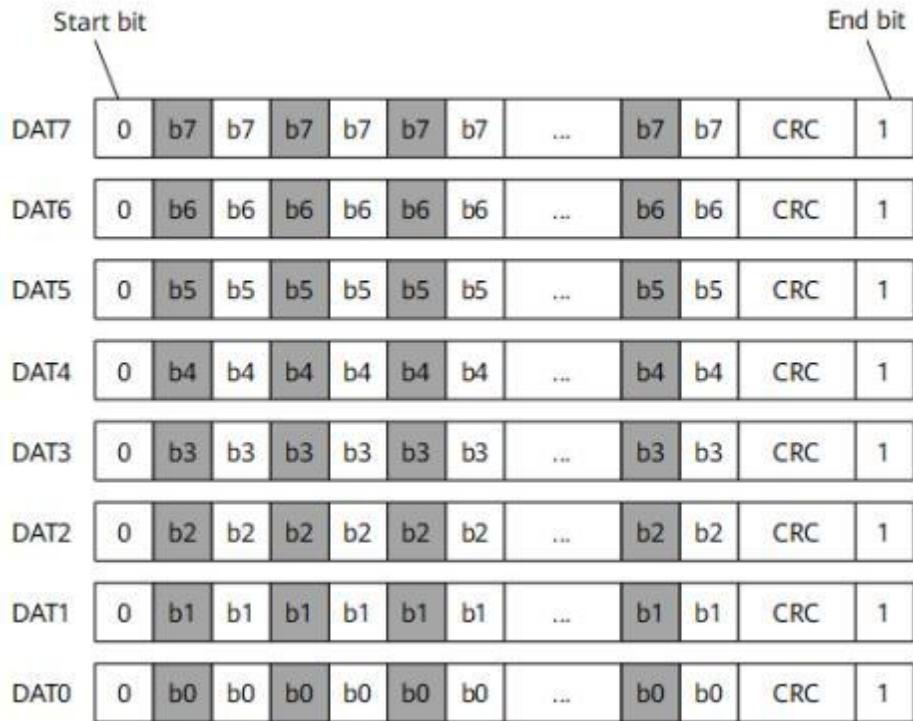
[Figure 12-16](#), [Figure 12-17](#), and [Figure 12-18](#) show the data transfer format in 1-bit, 4-bit, and 8-bit mode (eMMC only), respectively.

**Figure 12-16** Data transfer format in 1-bit mode



**Figure 12-17** Data transfer format in 4-bit mode



**Figure 12-18** Data transfer format in 8-bit mode

## Speed Modes Supported by SD 2.0

The controller supports SD 2.0. [Table 12-10](#) describes the transfer modes.

**Table 12-10** Transfer modes

Mode	Input Clock (MHz)	Card-Side Clock	Maximum Bit Width	Voltage
HS	50 MHz	50 MHz	4 bit	3.3 V
DS	25 MHz	25 MHz	4 bit	3.3 V

## Speed Modes Supported by SDIO 2.0

The controller supports SDIO 2.0. [Table 12-11](#) shows the transmission modes supported by the controller in SD mode.

**Table 12-11** Transmission modes

Mode	Interface Clock	Maximum Bit Width	Voltage
HS	50 MHz	4 bits	3.3 V



Mode	Interface Clock	Maximum Bit Width	Voltage
DS	25 MHz	4 bits	3.3 V

## Speed Modes and Voltages Supported by eMMC

Table 12-12 shows the transmission modes supported by the controller in eMMC mode.

**Table 12-12** Transmission modes

Mode	Interface Clock	Maximum Bit Width	Voltage
HS400	148.5 MHz	8 bits	1.8 V
HS200	148.5 MHz	8 bits	1.8 V
HS_SDR	50 MHz	8 bits	1.8 V/3.3 V
Backwards Compatibility with legacy MMC card	25 MHz	8 bits	1.8 V/3.3 V

## Application Notes

### NOTE

For details about clock reset registers, see section 3.2.5 "Register Description."

## Soft Reset

When the MMC fails to return to the idle state due to data transmission exceptions:

Write 1 to PERI\_CRG3440/PERI\_CRG3504 bit [16] to soft reset the SDIO0/SDIO1 module.

Before using the MMC, you are advised to soft-reset the MMC.

## Interface Clock Configuration

Clock frequencies vary according to the eMMC/SD/SDIO device complying with different protocols and the status of the eMMC/SD/SDIO device.

SDIO0/SDIO1 is configured by using PERI\_CRG3440/PERI\_CRG3504 bit[26:24].

Before changing the clock frequency of an eMMC/SD/SDIO, ensure that no data or command is being transferred. In addition, to avoid the occurrence of glitches in the output clock of the eMMC/SD/SDIO, you need to perform the following steps:



**Step 1** Disable the interface clock.

Set [CLK\\_CTRL\\_R](#)bit[2] to **0**.

**Step 2** Set the frequency of the interface clock.

Select the interface clock frequency by configuring PERI\_CRG3440/PERI\_CRG3504 bit[26:24].

**Step 3** Enable the interface clock again.

Set [CLK\\_CTRL\\_R](#)bit[2] to **1**.

----**End**

## MMC Initialization

Before commands and data are exchanged between a card and the controller, the MMC controller needs to be initialized. Perform the following steps:

**Step 1** Clear interrupts. Set [NORMAL\\_INT\\_STAT\\_R](#)bit[15:0] and [ERROR\\_INT\\_STAT\\_R](#)bit[15:0] to 1 to clear raw interrupt status bits.

**Step 2** Disable interrupts. Set [NORMAL\\_INT\\_SIGNAL\\_EN\\_R](#)bit[15:0] and [ERROR\\_INT\\_SIGNAL\\_EN\\_R](#)bit[15:0] to 0.

**Step 3** Set timeout count by configuring [TOUT\\_CTRL\\_R](#)bit[3:0].

**Step 4** If an eMMC is used, set [EMMC\\_CTRL\\_R](#)bit[0] to 1. Otherwise, set it to 0.

**Step 5** Enable power\_on. set [PWR\\_CTRL\\_R](#)bit[0] to 1.

**Step 6** Enable clocks. Set [CLK\\_CTRL\\_R](#) bit[3], bit[2], and bit[0] to 1.

**Step 7** Enable interrupt status, and configure [NORMAL\\_INT\\_STAT\\_EN\\_R](#)bit[15:0] and [ERROR\\_INT\\_STAT\\_R](#)bit[15:0].

**Step 8** Enable interrupt signals, and configure [NORMAL\\_INT\\_SIGNAL\\_EN\\_R](#) bit[15:0] and [ERROR\\_INT\\_SIGNAL\\_EN\\_R](#)bit[15:0].

----**End**

After the preceding steps, the interface clock can be configured and commands can be transmitted to the card.

## Non-Data Transfer Command

A non-data transfer command is sent as follows:

**Step 1** Configure the corresponding commands in [ARGUMENT\\_R](#).

**Step 2** Configure the corresponding command parameters in [XFER\\_MODE\\_R](#) and [CMD\\_R](#).

**Step 3** Wait until the MMC controller runs the command.



- Step 4** Check whether a command completion interrupt is generated by checking [NORMAL\\_INT\\_STAT\\_R](#)bit[0].
- Step 5** Check whether an exception interrupt is generated by checking [ERROR\\_INT\\_STAT\\_R](#) bit[15:0]. Read the response value when necessary.
- End

## Reading a Single Data Block or Multiple Data Blocks

To read a single data block or multiple data blocks, perform the following steps:

- Step 1** Clear interrupts in [NORMAL\\_INT\\_STAT\\_R](#) and [ERROR\\_INT\\_STAT\\_R](#).
- Step 2** Write the block size to [BLOCKSIZE\\_R](#) and [BLOCKCOUNT\\_R](#)bit[11:0].
- Step 3** Write the number of blocks to [SDMASA\\_R](#).
- Step 4** Write the start address for reading data to [ARGUMENT\\_R](#).
- Step 5** Configure [XFER\\_MODE\\_R](#) and [CMD\\_R](#) to set data transfer parameters.

For the eMMC/SD, read a single data block by running CMD17, or read multiple data blocks by running CMD18; for the SDIO card, read data block(s) by running CMD53.

The MMC controller starts to run commands after [CMD\\_R](#) is written. After commands are transferred to the bus, the cmd\_done interrupt is generated.

- Step 6** Read data in non\_DMA mode. Check [NORMAL\\_INT\\_STAT\\_R](#)bit[5]. If the value is 1, read the data in the FIFO from [BUF\\_DATA\\_R](#) so that the MMC can receive subsequent data. In addition, check data error interrupts by checking [ERROR\\_INT\\_STAT\\_R](#)bit[4:6]. In this case, the program can send a stop command to stop the data transfer.
- Step 7** If [NORMAL\\_INT\\_STAT\\_R](#)bit[1] is 1, data transfer is complete. In this case, read the remaining data in the FIFO by reading [BUF\\_DATA\\_R](#).
- End

## Writing a Single Data Block or Multiple Data Blocks

To write a single data block or multiple data blocks, perform the following steps:

- Step 1** Clear interrupts in [NORMAL\\_INT\\_STAT\\_R](#) and [ERROR\\_INT\\_STAT\\_R](#).
- Step 2** Write the block size to [BLOCKSIZE\\_R](#) and [BLOCKCOUNT\\_R](#)bit[11:0].
- Step 3** Write the number of blocks to [SDMASA\\_R](#).
- Step 4** Write the start address for writing data to [ARGUMENT\\_R](#).
- Step 5** Configure [XFER\\_MODE\\_R](#) and [CMD\\_R](#) to set data transfer parameters.



For the eMMC, read a single data block by running CMD24, or read multiple data blocks by running CMD25; for the SDIO card, read data block(s) by running CMD53.

- Step 6** Write data in non\_DMA mode. Write data to the FIFO, that is, write [BUF\\_DATA\\_R](#).
- Step 7** Check [NORMAL\\_INT\\_STAT\\_R](#) bit[4]. If the value is 1, write data that is to be filled in the FIFO to [BUF\\_DATA\\_R](#) so that the MMC can transfer subsequent data. In addition, check data error interrupts by checking [ERROR\\_INT\\_STAT\\_R](#) bit[4:6]. In this case, the program can send a stop command to stop the data transfer.
- Step 8** If [NORMAL\\_INT\\_STAT\\_R](#) bit[1] is 1, the data transfer is complete.  
----End

### 12.3.4 Register Summary

[Table 12-13](#) describes the MMC registers.

**Table 12-13** Summary of MMC registers (SDIO0/SDIO1 base address:  
0x1003\_0000/0x1004\_0000)

Offset Address	Name	Description	Page
0x0000	SDMASA_R	SDMA system address register	<a href="#">12-63</a>
0x0004	BLOCKSIZE_R	Block size register	<a href="#">12-63</a>
0x0006	BLOCKCOUNT_R	Block count register	<a href="#">12-64</a>
0x0008	ARGUMENT_R	SD card or eMMC command register	<a href="#">12-64</a>
0x000c	XFER_MODE_R	Transfer mode register	<a href="#">12-64</a>
0x000e	CMD_R	Command register	<a href="#">12-65</a>
0x0010	RESP01_R	RESP 0/1 register	<a href="#">12-66</a>
0x0014	RESP23_R	RESP 2/3 register	<a href="#">12-67</a>
0x0018	RESP45_R	RESP 4/5 register	<a href="#">12-67</a>
0x001c	RESP67_R	RESP 6/7 register	<a href="#">12-67</a>
0x0020	BUF_DATA_R	Buffer data register	<a href="#">12-67</a>
0x0024	PSTATE_R	Current state register	<a href="#">12-68</a>
0x0028	HOST_CTRL1_R	Host control 1 register	<a href="#">12-69</a>
0x0029	PWR_CTRL_R	Power-on control register	<a href="#">12-70</a>
0x002c	CLK_CTRL_R	Clock control register	<a href="#">12-70</a>



Offset Address	Name	Description	Page
0x002e	TOUT_CTRL_R	Timeout control register	<a href="#">12-71</a>
0x002f	SW_RST_R	Soft reset control register	<a href="#">12-71</a>
0x0030	NORMAL_INT_STAT_R	Normal interrupt status register	<a href="#">12-72</a>
0x0032	ERROR_INT_STAT_R	Error interrupt status register	<a href="#">12-73</a>
0x0034	NORMAL_INT_STAT_E_N_R	Normal interrupt status enable register	<a href="#">12-75</a>
0x0036	ERROR_INT_STAT_EN_R	Error interrupt status enable register	<a href="#">12-76</a>
0x0038	NORMAL_INT_SIGNAL_EN_R	Normal interrupt signal enable register	<a href="#">12-78</a>
0x003a	ERROR_INT_SIGNAL_EN_R	Error interrupt signal enable register	<a href="#">12-80</a>
0x003c	AUTO_CMD_STAT_R	Auto command and status register	<a href="#">12-81</a>
0x003e	HOST_CTRL2_R	Host control 2 register	<a href="#">12-82</a>
0x0040	CAPABILITIES1_R	Capabilities 1 register	<a href="#">12-84</a>
0x0044	CAPABILITIES2_R	Capabilities 2 register	<a href="#">12-86</a>
0x0048	CURR_CAPBILITIES1_R	Current capabilities 1 register	<a href="#">12-87</a>
0x0054	ADMA_ERR_STAT_R	ADMA error status register	<a href="#">12-87</a>
0x0058	ADMA_SA_LOW_R	Lower 32-bit register for the ADMA system address	<a href="#">12-88</a>
0x005c	ADMA_SA_HIGH_R	Upper 32-bit register for the ADMA system address	<a href="#">12-88</a>
0x0078	ADMA_ID_LOW_R	Lower 32-bit register for the ADMA integrated descriptor address	<a href="#">12-88</a>
0x007c	ADMA_ID_HIGH_R	Upper 32-bit register for the ADMA integrated descriptor address	<a href="#">12-89</a>
0x0500	MSHC_VER_ID_R	Controller version ID register	<a href="#">12-89</a>
0x0504	MSHC_VER_TYPE_R	Controller version type register	<a href="#">12-89</a>
0x0510	MBIU_CTRL_R	MBIU control register	<a href="#">12-89</a>



Offset Address	Name	Description	Page
0x052c	EMMC_CTRL_R	eMMC control register	<a href="#">12-91</a>
0x052e	BOOT_CTRL_R	Boot control register	<a href="#">12-91</a>
0x0534	GP_OUT_R	eMMC reset and GPIO register	<a href="#">12-92</a>
0x0540	AT_CTRL_R	Tuning control register	<a href="#">12-92</a>
0x0544	AT_STAT_R	Tuning status register	<a href="#">12-93</a>

## 12.3.5 Register Description

### **SDMASA\_R**

SDMASA\_R is an SDMA system address register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	blockcnt_sdmasa	When <b>Host Version Enable</b> is set to <b>0</b> , the register indicates the SDMA system address. When <b>Host Version Enable</b> is set to <b>1</b> , the register indicates the block count. The value <b>1</b> indicates one block.	0x00000000

### **BLOCKSIZE\_R**

BLOCKSIZE\_R is a block size register.

Offset Address: 0x0004 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	-	reserved	Reserved	0x0
[14:12]	RW	sdma_buf_bdary	Boundary value of an SDMA data block 000: 4 KB 001: 8 KB ... 111: 512 KB	0x0



Bits	Access	Name	Description	Reset
[11:0]	RW	xfer_block_size	Transferred block size (in bytes) Maximum value: <b>0x800</b>	0x000

## BLOCKCOUNT\_R

BLOCKCOUNT\_R is a block count register.

Offset Address: 0x0006 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	block_cnt	Block count	0x0000

## ARGUMENT\_R

ARGUMENT\_R is an SD card or eMMC command content register.

Offset Address: 0x0008 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	argument	Content value of the 32-bit command	0x00000000

## XFER\_MODE\_R

XFER\_MODE\_R is a transfer mode command register.

Offset Address: 0x000c Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:9]	-	reserved	Reserved	0x00
[8]	RW	resp_int_disable	Response interrupt disable 0: enabled 1: disabled	0x0
[7]	RW	resp_err_chk_enable	Response error check enable 0: disabled 1: enabled	0x0



Bits	Access	Name	Description	Reset
[6]	RW	resp_type	Response R1/R5 type 0: R1 1: R5	0x0
[5]	RW	multi_blk_sel	Multiple/single block select 0: single block 1: multiple blocks	0x0
[4]	RW	data_xfer_dir	Data transfer direction 0: from the controller to the card 1: from the card to the controller	0x0
[3:2]	RW	auto_cmd_enable	Auto command enable 00: disabled 01: auto CMD12 enabled 10: auto CMD23 enabled 11: auto CMD auto select (The value is fixed at <b>00</b> in the SDIO.)	0x0
[1]	RW	block_count_enable	Block count enable 0: disabled 1: enabled	0x0
[0]	RW	dma_enable	DMA enable 0: No data is transmitted or non-DMA data is transmitted. 1: DMA data is transmitted.	0x0

## CMD\_R

CMD\_R is a transfer mode command register.

Offset Address: 0x000e

Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:14]	-	reserved	Reserved	0x0
[13:8]	RW	cmd_index	Command index	0x00
[7:6]	RW	cmd_type	Command type 00: normal command	0x0



Bits	Access	Name	Description	Reset
			01: suspend command 10: resume command 11: abort command	
[5]	RW	data_present_sel	Data transmission select 0: No data is transmitted. 1: Data is transmitted.	0x0
[4]	RW	cmd_idx_chk_enable	Command ID check enable 0: disabled 1: enabled	0x0
[3]	RW	cmd_crc_chk_enable	Command cyclic redundancy check (CRC) enable 0: disabled 1: enabled	0x0
[2]	RW	sub_cmd_flag	Subcommand flag 0: main 1: subcommand	0x0
[1:0]	RW	resp_type_select	Response type 00: no response 01: response length 136 10: response length 48 11: response length 48 check	0x0

## RESP01\_R

RESP01\_R is a RESP 0/1 register.

Offset Address: 0x0010 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	resp01	RESP 0/1 register. This register stores the content of the response bits [39:8] of the send command of the controller.	0x00000000



## RESP23\_R

RESP23\_R is a RESP 2/3 register.

Offset Address: 0x0014 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	resp23	RESP 2/3 register. This register stores the content of the response bits [71:40] of the send command of the controller.	0x00000000

## RESP45\_R

RESP45\_R is a RESP 4/5 register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	resp45	RESP 4/5 register. This register stores the content of the response bits [103:72] of the send command of the controller.	0x00000000

## RESP67\_R

RESP67\_R is a RESP 6/7 register.

Offset Address: 0x001c Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	resp67	RESP 6/7 register. This register stores the content of the response bits [135:104] of the send command of the controller.	0x00000000

## BUF\_DATA\_R

BUF\_DATA\_R is a buffer data register.

Offset Address: 0x0020 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	buf_data	In slave mode, the buffer used to store	0x00000000



			the RX data and TX data in the controller writes data to the card by continuously writing data to this register and read data from the card by continuously reading data in this register.	
--	--	--	--	--

## PSTATE\_R

PSTATE\_R is a current state register.

Offset Address: 0x0024 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27]	RO	cmd_issue_err	Command issuing error 0: There is no command issuing error. 1: The command cannot be issued.	0x0
[26:25]	-	reserved	Reserved	0x0
[24]	RO	cmd_line_lvl	Level state the command pin 0: low 1: high	0x0
[23:20]	RO	sd_dat_in[3:0]	State of the data[3:0] pin, corresponding to each bit 0 The data pin is in low level. 1: The data pin is in high level.	0x0
[19:12]	-	reserved	Reserved	0x0
[11]	RO	buf_rd_enable	Buffer read enable 0: disabled 1: enabled	0x0
[10]	RO	buf_wr_enable	Buffer write enable 0: disabled 1: enabled	0x0
[9]	RO	Rd_xfer_active	Read transfer active 0: No data is transferred (idle state). 1: Data is being read.	0x0
[8]	RO	Wr_xfer_active	Write transfer active	0x0



Bits	Access	Name	Description	Reset
			0: No data is transferred (idle state). 1: Data is being written.	
[7:4]	RO	sd_dat_in[7:4]	State of the data[7:4] pin, corresponding to each bit 0: The data pin is in low level. 1: The data pin is in high level.	0x0
[3]	-	reserved	Reserved	0x0
[2]	RO	Dat_line_active	Data line active 0: No data is transmitted (idle state). 1: Data is being transmitted.	0x0
[1]	RO	cmd_inhibit_dat	Flag indicating whether the command with data is active 0: No data is transmitted (idle state). Commands with data can be transmitted. 1: Data line is transmitting data or the read operation is being valid.	0x0
[0]	RO	cmd_inhibit	Data line active 0: idle state. The controller can send commands. 1: busy state. The controller cannot send commands.	0x0

## HOST\_CTRL1\_R

HOST\_CTRL1\_R is host control 1 register.

Offset Address: 0x0028    Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:6]	-	reserved	Reserved	0x0
[5]	RW	ext_data_xfer	Extended data bit width 0: See the dat_xfer_width register. 1: 8 bits	0x0
[5]	-	reserved	Reserved	0x0
[4:3]	RW	dma_sel	DMA select	0x0



Bits	Access	Name	Description	Reset
			<p>When <b>Host Version 4 Enable</b> in the HOST_CTRL2_R register is set to <b>1</b>:</p> <p>00: SDMA</p> <p>01: reserved</p> <p>10: ADMA2</p> <p>11: ADMA2 or ADMA3</p> <p>When <b>Host Version 4 Enable</b> in the HOST_CTRL2_R register is set to <b>0</b>:</p> <p>00: SDMA</p> <p>01: reserved</p> <p>10: 32-bit address ADMA2</p> <p>11: 64-bit address ADMA2</p>	
[2]	RW	high_speed_en	High speed enable 0: normal speed 1: high speed	0x0
[1]	RW	dat_xfer_width	TX data bit width 0: 1 bit 1: 4 bits	0x0
[0]	-	reserved	Reserved	0x0

## PWR\_CTRL\_R

PWR\_CTRL\_R is a host basic configuration control register.

Offset Address: 0x0029 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:1]	-	reserved	Reserved	0x00
[0]	RW	sd_bus_pwr_vdd1	VDD2 power enable 0: powered off 1: powered on	0x0

## CLK\_CTRL\_R

CLK\_CTRL\_R is a clock control register.



Offset Address: 0x002c Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x00
[3]	RW	pll_enable	PLL enable 0: The PLL is in low power mode. 1: The PLL is enabled.	0x0
[2]	RW	sd_clk_en	SDIO/eMMC clock enable 0: disabled 1: enabled	0x0
[1]	RO	internal_clk_stab_le	Internal clock state 0: unstable 1: stable	0x0
[0]	RW	internal_clk_en	Internal clock enable 0: disabled 1: enabled	0x1

## TOUT\_CTRL\_R

TOUT\_CTRL\_R is a timeout control register.

Offset Address: 0x002e Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:4]	-	reserved	Reserved	0x0
[3:0]	RW	tout_cnt	Data timeout count 0x0: TMCLK x 2^13 ... 0xe: TMCLK x 2^27 Other values: reserved	0x0

## SW\_RST\_R

SW\_RST\_R is a soft reset control register.

Offset Address: 0x002f Total Reset Value: 0x00



Bits	Access	Name	Description	Reset
[7:3]	-	reserved	Reserved	0x00
[2]	RW	sw_RST_dat	Data line request for soft reset 0: reset deasserted 1: reset	0x0
[1]	RW	sw_RST_cmd	Command line request for soft reset 0: reset deasserted 1: reset	0x0
[0]	RW	sw_RST_all	Controller request for soft reset 0: reset deasserted 1: reset	0x0

## NORMAL\_INT\_STAT\_R

NORMAL\_INT\_STAT\_R is a normal interrupt status register.

Offset Address: 0x0030 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	RO	err_interrupt	Error interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[14]	WC	cqe_event	CMDQ event interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[13]	RO	fx_event	FX event interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[12]	-	reserved	Reserved	0x0
[11]	RO	int_c	INT_C interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[10]	RO	int_b	INT_B interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
[9]	RO	int_a	INT_A interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	card_interrupt	Card interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7:6]	-	reserved	Reserved	0x0
[5]	WC	buf_rd_ready	Ready interrupt status for buffer read 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	WC	buf_wr_ready	Ready interrupt status for buffer write 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	WC	dma_interrupt	DMA interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	WC	bgap_event	Block gap event interrupt due to a request stop 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	WC	xfer_comlete	Read/write transfer completion interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	WC	cmd_complete	Command completion interrupt status 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## ERROR\_INT\_STAT\_R

ERROR\_INT\_STAT\_R is an error interrupt status register.

Offset Address: 0x0032 Total Reset Value: 0x0000



Bits	Access	Name	Description	Reset
[15:13]	-	reserved	Reserved	0x0
[12]	WC	boot_ack_err	Boot acknowledgement error interrupt, used only in eMMC mode 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[11]	WC	resp_err	Response error interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[10]	-	reserved	Reserved	0x0
[9]	WC	adma_err	ADMA error interrupt, that is, response to a system bus error, invalid ADMA 2/3 descriptor, or invalid CQE descriptor. 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	WC	auto_cmd_err	Interrupt status of an auto command error 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7]	WC	cur_lmt_err	Interrupt status of a current limit error 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	WC	data_end_bit_err	Interrupt status of a data end bit error 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[5]	WC	data_crc_err	Interrupt status of a data CRC error 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	WC	data_tout_err	Interrupt status of a data timeout error 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	WC	cmd_idx_err	Interrupt status of a command index error 0: No interrupt is generated. 1: An interrupt is generated.	0x0



Bits	Access	Name	Description	Reset
[2]	WC	cmd_end_bit_err	Interrupt status of a command end bit error 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	WC	cmd_crc_err	Interrupt status of a command CRC error 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	WC	cmd_tout_err	Interrupt status of a command timeout error 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## NORMAL\_INT\_STAT\_EN\_R

NORMAL\_INT\_STAT\_EN\_R is a normal interrupt status enable register.

Offset Address: 0x0034      Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	-	reserved	Reserved	0x0
[14]	RW	cqe_event_stat_en	CMDQ event interrupt status enable 0: disabled 1: enabled	0x0
[13]	RW	fx_event_stat_en	FX event interrupt status enable 0: disabled 1: enabled	0x0
[12]	-	reserved	Reserved	0x0
[11]	RW	int_c_stat_en	INT_C interrupt status enable 0: disabled 1: enabled	0x0
[10]	RW	int_b_stat_en	INT_B interrupt status enable 0: disabled 1: enabled	0x0
[9]	RW	int_a_stat_en	INT_A interrupt status enable	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[8]	RW	card_interrupt_stat_en	Card interrupt status enable 0: disabled 1: enabled	0x0
[7:6]	-	reserved	Reserved	0x0
[5]	RW	buf_rd_ready_stat_en	Ready interrupt enable for buffer read 0: disabled 1: enabled	0x0
[4]	RW	buf_wr_ready_stat_en	Ready interrupt enable for buffer write 0: disabled 1: enabled	0x0
[3]	RW	dma_interrupt_stat_en	DMA interrupt enable 0: disabled 1: enabled	0x0
[2]	RW	bgap_event_stat_en	Block gap event interrupt enable due to request stop 0: disabled 1: enabled	0x0
[1]	RW	xfer_complete_stat_en	Interrupt status enable for read/write transfer completion 0: disabled 1: enabled	0x0
[0]	RW	cmd_complete_stat_en	Interrupt status enable for command completion 0: disabled 1: enabled	0x0

## ERROR\_INT\_STAT\_EN\_R

ERROR\_INT\_STAT\_EN\_R is an error interrupt status enable register.

Offset Address: 0x0036 Total Reset Value: 0x0000



Bits	Access	Name	Description	Reset
[15:13]	-	reserved	Reserved	0x0
[12]	RW	boot_ack_err_stat_en	Interrupt status enable for a boot acknowledgement error, used only in eMMC mode 0: disabled 1: enabled	0x0
[11]	RW	resp_err_stat_en	Interrupt status enable for a response error 0: disabled 1: enabled	0x0
[10]	-	reserved	Reserved	0x0
[9]	RW	adma_err_stat_en	ADMA error interrupt status enable System bus error response, invalid ADMA2/3 descriptor, or invalid CQE descriptor 0: disabled 1: enabled	0x0
[8]	RW	auto_cmd_err_stat_en	Interrupt status enable for an auto command error 0: disabled 1: enabled	0x0
[7]	RW	cur_lmt_err_stat_en	Interrupt status enable for a current limit error 0: disabled 1: enabled	0x0
[6]	RW	data_end_bit_err_stat_en	Interrupt status enable for a data end bit error 0: disabled 1: enabled	0x0
[5]	RW	data_crc_err_stat_en	Interrupt status enable for a data CRC error 0: disabled 1: enabled	0x0
[4]	RW	data_tout_err_stat_en	Interrupt status enable for a data timeout error	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[3]	RW	cmd_idx_err_stat_en	Interrupt status enable for a command index error 0: disabled 1: enabled	0x0
[2]	RW	cmd_end_bit_err_stat_en	Interrupt status enable for a command end bit error 0: disabled 1: enabled	0x0
[1]	RW	cmd_crc_err_stat_en	Interrupt status enable for a command CRC error 0: disabled 1: enabled	0x0
[0]	RW	cmd_tout_err_stat_en	Interrupt status enable for a command timeout error 0: disabled 1: enabled	0x0

## NORMAL\_INT\_SIGNAL\_EN\_R

NORMAL\_INT\_SIGNAL\_EN\_R is a normal interrupt signal enable register.

Offset Address: 0x0038 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	-	reserved	Reserved	0x0
[14]	RW	cqe_event_signal_en	CMDQ event interrupt signal enable 0: disabled 1: enabled	0x0
[13]	RW	fx_event_signal_en	FX event interrupt signal enable 0: disabled 1: enabled	0x0
[12]	-	reserved	Reserved	0x0
[11]	RW	int_c_signal_en	INT_C interrupt signal enable	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[10]	RW	int_b_signal_en	INT_B interrupt signal enable 0: disabled 1: enabled	0x0
[9]	RW	int_a_signal_en	INT_A interrupt signal enable 0: disabled 1: enabled	0x0
[8]	RW	card_interrupt_sig_nal_en	Card interrupt signal enable 0: disabled 1: enabled	0x0
[7:6]	-	reserved	Reserved	0x0
[5]	RW	buf_rd_ready_sign al_en	Ready signal enable for buffer read 0: disabled 1: enabled	0x0
[4]	RW	buf_wr_ready_sign al_en	Ready signal enable for buffer write 0: disabled 1: enabled	0x0
[3]	RW	dma_interrupt_sig_nal_en	DMA interrupt signal enable 0: disabled 1: enabled	0x0
[2]	RW	bgap_event_signal_en	Block gap event interrupt signal enable due to request stop 0: disabled 1: enabled	0x0
[1]	RW	xfer_comlete_sign al_en	Interrupt signal enable for read/write transfer completion 0: disabled 1: enabled	0x0
[0]	RW	cmd_complete_sig_nal_en	Interrupt signal enable for command completion 0: disabled 1: enabled	0x0



## ERROR\_INT\_SIGNAL\_EN\_R

ERROR\_INT\_SIGNAL\_EN\_R is an error interrupt signal enable register.

Offset Address: 0x003a Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:13]	-	reserved	Reserved	0x0
[12]	RW	boot_ack_err_signal_en	Interrupt signal enable for a boot acknowledgement error, used only in eMMC mode 0: disabled 1: enabled	0x0
[11]	RW	resp_err_signal_en	Interrupt signal enable for a response error. 0: disabled 1: enabled	0x0
[10]	-	reserved	Reserved	0x0
[9]	RW	adma_err_signal_en	Interrupt signal enable for an ADMA error, that is, response to a system bus error, invalid ADMA 2/3 descriptor, or invalid CQE descriptor. 0: disabled 1: enabled	0x0
[8]	RW	auto_cmd_err_signal_en	Interrupt signal enable for an auto command error 0: disabled 1: enabled	0x0
[7]	RW	cur_lmt_err_signal_en	Interrupt signal enable for a current limit error 0: disabled 1: enabled	0x0
[6]	RW	data_end_bit_err_signal_en	Interrupt signal enable for a data end bit error 0: disabled 1: enabled	0x0
[5]	RW	data_crc_err_signa	Interrupt signal enable for a data CRC	0x0



Bits	Access	Name	Description	Reset
		I_en	error 0: disabled 1: enabled	
[4]	RW	data_tout_err_sign al_en	Interrupt signal enable for a data timeout error 0: disabled 1: enabled	0x0
[3]	RW	cmd_idx_err_signal _en	Interrupt signal enable for a command index error 0: disabled 1: enabled	0x0
[2]	RW	cmd_end_bit_err_si gnal_en	Interrupt signal enable for a command end bit error 0: disabled 1: enabled	0x0
[1]	RW	cmd_crc_err_signal _en	Interrupt signal enable for a command CRC error 0: disabled 1: enabled	0x0
[0]	RW	cmd_tout_err_sign al_en	Interrupt signal enable for a command timeout error 0: disabled 1: enabled	0x0

## AUTO\_CMD\_STAT\_R

AUTO\_CMD\_STAT\_R is an auto command status register.

Offset Address: 0x003c Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x00
[7]	RO	cmd_not_issued_a uto_cmd12	Subsequent command sending status when an auto CMD12 error occurs 0: No error occurs. 1: The CMD is not sent through the data	0x0



Bits	Access	Name	Description	Reset
			line.	
[6]	-	reserved	Reserved	0x0
[5]	RO	auto_cmd_resp_err	Auto CMD response status 0: No error occurs. 1: An error occurs.	0x0
[4]	RO	auto_cmd_idx_err	Auto CMD index status 0: No error occurs. 1: An error occurs.	0x0
[3]	RO	auto_cmd_endbit_err	Auto CMD end bit status 0: No error occurs. 1: An error occurs.	0x0
[2]	RO	auto_cmd_crc_err	Auto CMD CRC status 0: No error occurs. 1: An error occurs.	0x0
[1]	RO	auto_cmd_tout_err	Auto CMD timeout status 0: No error occurs. 1: An error occurs.	0x0
[0]	RO	auto_cmd12_not_exec	Auto CMD12 execution status 0: executed 1: not executed	0x0

## HOST\_CTRL2\_R

HOST\_CTRL2\_R is host control 2 register.

Offset Address: 0x003e Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15]	RW	preset_val_enable	Automatic select enable of the preset value 0: disabled. The SDCLK and drive capability are controlled by the driver. 1: enabled	0x0
[14]	RW	async_int_enable	Asynchronization interrupt enable	0x0



Bits	Access	Name	Description	Reset
			0: disabled 1: enabled	
[13]	-	reserved	Reserved	0x0
[12]	RW	host_ver4_enable	Controller version 4.0 enable 0: version 3.0 1: version 4.0	0x0
[11]	RW	cmd23_enable	CMD23 enable 0: disabled 1: enabled	0x0
[10]	RW	adma2_len_mode	ADMA2 length 0: 16-bit data length 1: 26-bit data length	0x0
[9:8]	-	reserved	Reserved	0x0
[7]	RW	sample_clk_sel	Sample clock select 0: fixed clock 1: tuned clock	0x0
[6]	RW	exec_tuning	Tuning execution. After the operation is complete, this register is automatically cleared and set to 0. 0: Tuning is not executed or has been executed. 1: Tuning is executed.	0x0
[5:4]	RW	drv_strength_sel	Drive capability select 00:TYPEB 01: TYPEA 10: TYPEC 11: TYPED	0x0
[3]	-	reserved	Reserved	0x0
[2:0]	RW	uhs_mode_sel	UHS mode 000: SDR12 001: SDR25 010: SDR50 011: SDR104 100: reserved	0x0



Bits	Access	Name	Description	Reset
			101: reserved 110: reserved 111: reserved eMMC mode 000: legacy 001: high speed SDR 010: reserved 011: HS200 100: high speed DDR 100: reserved 101: reserved 110: reserved 111: HS400	

## CAPABILITIES1\_R

CAPABILITIES1\_R is capabilities 1 register.

Offset Address: 0x0040 Total Reset Value: 0x3F6E\_0181

Bits	Access	Name	Description	Reset
[31:30]	RO	slot_type_r	Slot type 00: removable cardslot 01: embedded slot 10: shared bus slot, used in SD card mode 11: reserved	0x0
[29]	RO	async_int_support	Whether the synchronization interrupt is supported 0: not supported 1: supported	0x1
[28]	-	reserved	Reserved	0x1
[27]	-	reserved	Reserved	0x1
[26:24]	-	reserved	Reserved	0x7
[23]	RO	sus_res_support	Whether pausing and resuming are supported	0x0



Bits	Access	Name	Description	Reset
			0: not supported 1: supported	
[22]	RO	sdma_support	Whether the SDMA is supported 0: not supported 1: supported	0x1
[21]	RO	high_speed_support	Whether a high speed is supported 0: not supported 1: supported	0x1
[20]	-	reserved	Reserved	0x0
[19]	RO	adma2_support	Whether ADMA2 is supported 0: not supported 1: supported	0x1
[18]	RO	embedded_8_bit	Whether 8-bit embedded components are supported 0: not supported 1: supported	0x1
[17:16]	RO	max_blk_len	Maximum block length 00: 512 bytes 01: 1024 bytes 10: 2048 bytes 11: reserved	0x2
[15:8]	RO	base_clk_freq	Basic frequency of the SD clock 0x0: 1 MHz 0x3f: 63 MHz 0x40-0xff: not supported	0x01
[7]	RO	tout_clk_unit	Timeout clock unit 0: KHz 1: MHz	0x1
[6]	-	reserved	Reserved	0x0
[5:0]	RO	tout_clk_freq	Timeout frequency 0x1: 1 KHz/1 MHz 0x2: 2 KHz/2 MHz ...	0x01



Bits	Access	Name	Description	Reset
			0x3f: 63 KHz/63 MHz	

## CAPABILITIES2\_R

CAPABILITIES2\_R is capabilities 2 register.

Offset Address: 0x0044 Total Reset Value: 0x0800\_2077

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27]	RO	adma3_support	Whether ADMA3 is supported 0: not supported 1: supported	0x1
[26:16]	-	reserved	Reserved	0x000
[15:14]	RO	re_tuning_modes	Retuning mode 00: mode 1, timer 01: mode 2, timer, and retuning requests 10: mode 3, auto retuning timer, and retuning requests 11: reserved	0x0
[13]	RO	use_tuning_ser50	Whether tuning is performed for SDR50 0: no 1: yes	0x1
[12]	-	reserved	Reserved	0x0
[11:8]	RO	retune_cnt	Retuning count 0x1: 1 second 0x3: 4 seconds Other values: reserved	0x0
[7]	-	reserved	Reserved	0x0
[6]	RO	drv_typed	Whether TYPED is supported 0: not supported 1: supported	0x1
[5]	RO	drv_typec	Whether TYPEC is supported 0: not supported	0x1



Bits	Access	Name	Description	Reset
			1: supported	
[4]	RO	drv_typea	Whether TYPEA is supported 0: not supported 1: supported	0x1
[3]	RO	uhs2_support	Whether UHS2 is supported 0: not supported 1: supported	0x0
[2]	-	reserved	Reserved	0x1
[1]	RO	sdr104_support	Whether SDR104 is supported 0: not supported 1: supported	0x1
[0]	RO	sdr50_support	Whether SDR50 is supported 0: not supported 1: supported	0x1

## CURR\_CAPBILITIES1\_R

CURR\_CAPBILITIES1\_R is current capabilities 1 register.

Offset Address: 0x0048 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:16]	RO	max_cur_18v	Current value at 1.8 V	0x00
[15:8]	RO	max_cur_30v	Current value at 3.0 V	0x00
[7:0]	RO	mac_cur_33v	Current value at 3.3 V	0x00

## ADMA\_ERR\_STAT\_R

ADMA\_ERR\_STAT\_R is an ADMA error status register.

Offset Address: 0x0054 Total Reset Value: 0x00



Bits	Access	Name	Description	Reset
[7:3]	-	reserved	Reserved	0x00
[2]	RO	adma_len_err	ADMA length mismatch error status 0: No error occurs. 1: An error occurs.	0x0
[1:0]	RO	adma_err_states	ADMA error status 00: ST_STOP 01: ST_FDS 10: ST_UNUSED 11: ST_TFR	0x0

### **ADMA\_SA\_LOW\_R**

ADMA\_SA\_LOW\_R is a lower 32-bit register for the ADMA system address.

Offset Address: 0x0058 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	adma_sa_low	Lower 32 bits of the ADMA system address	0x00000000

### **ADMA\_SA\_HIGH\_R**

ADMA\_SA\_HIGH\_R is an upper 32-bit register for the ADMA system address.

Offset Address: 0x005c Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	adma_sa_high	Upper 32 bits of the ADMA system address	0x00000000

### **ADMA\_ID\_LOW\_R**

ADMA\_ID\_LOW\_R is a lower 32-bit register for the ADMA integrated descriptor address.

Offset Address: 0x0078 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:0]	RW	adma_id_low	Lower 32 bits of the ADMA integrated descriptor address	0x00000000

### **ADMA\_ID\_HIGH\_R**

ADMA\_ID\_HIGH\_R is an upper 32-bit register for the ADMA integrated descriptor address.

Offset Address: 0x007c Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	adma_id_high	Upper 32 bits of the ADMA integrated descriptor address	0x00000000

### **MSHC\_VER\_ID\_R**

MSHC\_VER\_ID\_R is a controller version ID register.

Offset Address: 0x0500 Total Reset Value: 0x3137\_302A

Bits	Access	Name	Description	Reset
[31:0]	RO	mshc_ver_id	Controller version ID register	0x3137302A

### **MSHC\_VER\_TYPE\_R**

MSHC\_VER\_TYPE\_R is a controller version type register.

Offset Address: 0x0504 Total Reset Value: 0x6761\_2A2A

Bits	Access	Name	Description	Reset
[31:0]	RO	mshc_ver_id	Controller version type register	0x67612A2A

### **MBIU\_CTRL\_R**

MBIU\_CTRL\_R is an MBIU control register.

Offset Address: 0x0510 Total Reset Value 0x0707\_000F



Bits	Access	Name	Description	Reset
[31:27]	-	reserved	Reserved	0x000
[26:24]	RW	gm_write_osrc_lmt	Outstanding value of the bus write operation  The value is the register value plus 1.	0x7
[23:19]	-	reserved	Reserved	0x00
[18:16]	RW	gm_read_osrc_lmt	Outstanding value of the bus read operation.  The value is the register value plus 1.	0x7
[15:10]	-	reserved	Reserved	0x00
[9]	RW	gm_onerequestonly	Validity enable for outstanding configurations  0: outstanding valid based on gm_write_osrc_lmt /read_osrc_lmt 1: outstanding fixed to be 1	0x0
[8]	RW	gm_axi_1k_en	AXI bus cross 1 KB boundary read/write enable  0: disabled 1: enabled	0x0
[7:4]	-	reserved	Reserved	0x0
[3]	RW	gm_enburst16	16-burst enable  0: disabled 1: enabled	0x1
[2]	RW	gm_enburst8	8-burst enable  0: disabled 1: enabled	0x1
[1]	RW	gm_enburst4	4-burst enable  0: disabled 1: enabled	0x1
[0]	RW	gm_enburst_undef	Validity enable for burst configurations  0: A fixed burst is generated based on gm_enburst4/8/16. 1: A corresponding burst is generated based on the actual data length.	0x1



## EMMC\_CTRL\_R

EMMC\_CTRL\_R is an eMMC control register.

Offset Address: 0x052c Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:9]	-	reserved	Reserved	0x00
[8]	RW	enh_strobe_enable	Enhanced strobe enable This field indicates that the controller uses data strobe to sample the CMD line in HS400 mode. 0: In HS400 mode, cclk_rx is used to sample the CMD line. 1: In HS400 mode, data strobe is used to sample the CMD line.	0x0
[7:2]	-	reserved	Reserved	0x00
[1]	RW	disable_data_crc_ck	Data CRC disable 0: Data CRC is enabled. 1: Data CRC is disabled.	0x00
[0]	RW	card_is_emmc	Type of the connected card 0: non-eMMC card 1: eMMC card	0x0

## BOOT\_CTRL\_R

BOOT\_CTRL\_R is a boot control register.

Offset Address: 0x052e Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:12]	RW	boot_tout_cnt	Timeout count of boot acknowledgement 0x0: TMCLK x $2^{13}$ 0x1: TMCLK x $2^{14}$ ... 0xE: TMCLK x $2^{27}$ Other values: reserved	0x00
[11:9]	-	reserved	Reserved	0x0



Bits	Access	Name	Description	Reset
[8]	RW	boot_ack_enable	Boot acknowledgement enable 0: disabled 1: enabled	0x0
[7]	WO	validate_boot	Validity enable for forcible boot 0: There is no effect. 1: The boot enable is valid.	0x0
[6:1]	-	reserved	Reserved	0x00
[0]	RW	man_boot_en	Forcible boot enable. This bit must be configured together with validate_boot. It is automatically cleared after the boot transfer is complete. 0: There is no effect. 1: The boot enable is valid.	0x0

## GP\_OUT\_R

GP\_OUT\_R is an eMMC reset and GPIO register.

Offset Address: 0x0534 Total Reset Value: 0x0000\_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x0000000
[0]	RW	emmc_RST_N	eMMC card reset signal 0: reset 1: not reset	0x1

## AT\_CTRL\_R

AT\_CTRL\_R is a tuning control register.

Offset Address: 0x0540 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x0000000
[4]	RW	sw_tuning_en	Software configuration tuning enable 0: disabled	0x00



Bits	Access	Name	Description	Reset
			1: enabled	
[3:0]	-	reserved	Reserved	0x0

## AT\_STAT\_R

AT\_STAT\_R is a tuning status register.

Offset Address: 0x0544 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x0000000
[7:0]	RW	sw_ph_sel_code	<p>Phase value of the software configuration 0x0: 0° 0x1: 11.25° 0x2: 22.5° ... 0x1F: 348.75° Other values: reserved</p> <p> <b>NOTE</b> The SD/SDIO/eMMC in DS/HS mode supports only the following phases: 0x0, 0x4, 0x8, 0xC, 0x10, 0x14, 0x18, and 0x1C.</p>	0x0

## 12.4 SPI

### 12.4.1 Overview

The serial peripheral interface (SPI) controller implements serial-to-parallel conversion and parallel-to-serial conversion, and serves as a master to communicate with peripherals in synchronous and serial modes. The SPI controller supports three types of peripheral interfaces including the Motorola SPI, TI synchronous serial interface, and National Semiconductor MicroWire interface.

### 12.4.2 Features



## NOTICE

- The SoC provides two SPIs.
- The working reference clock is 100 MHz. The maximum SPI\_CLK output in SPI0/1 master mode is 25 MHz, and the maximum SPI\_CLK input in SPI0/1 slave mode is 8.25 MHz.

The SPI controller has the following features:

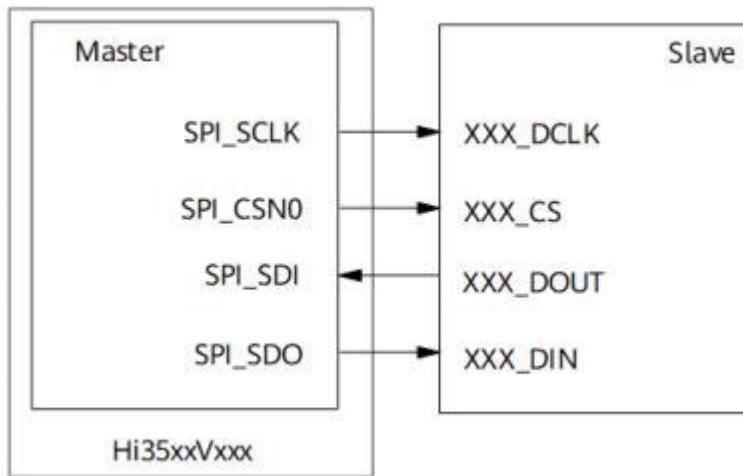
- Programmable frequency of the interface clock
- All the two groups of SPI interfaces support the master mode and slave mode.
- SPI0 supports dual CSs, while SPI1 only supports single CS. The polarity of the CS signal is configurable.
- The width of the RX FIFO is 16 bits and the depth is 256.
- The width of the TX FIFO is 16 bits and the depth is 256.
- Programmable serial data frame length: 4 bits to 16 bits
- Internal loopback test mode
- Direct memory access (DMA)
- Three types of peripheral interfaces (SPI, MicroWire, and TI synchronous serial interface), supporting single frame and consecutive frames
- SPI in full-duplex mode and configurable clock polarity and phase
- MicroWire in half-duplex mode
- TI synchronous serial interface in full-duplex mode

### 12.4.3 Function Description

#### Typical Application

Figure 12-19 shows the application block diagram when the SPI is connected to a slave device. The default CS pin SPI\_CSNO is used.

**Figure 12-19 Application block diagram when the SPI is connected to a single slave device**



#### 12.4.4 Peripheral Bus Timings

The meanings of the abbreviations and acronyms in [Figure 12-20](#) to [Figure 12-27](#) are as follows:

- MSB: most significant bit
- LSB: least significant bit
- Q: Q is an undefined signal

#### Motorola SPI

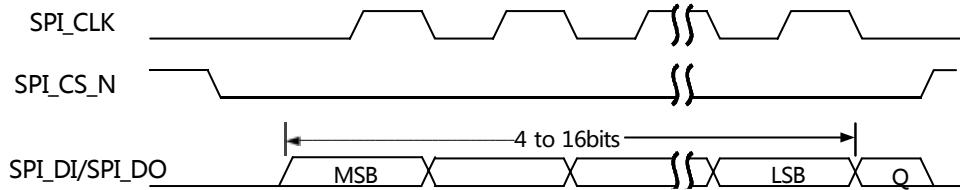
##### NOTE

SPO indicates the polarity of SPICLKOUT and SPH indicates the phase of SPICLKOUT. They correspond to SPICR0 bit[7: 6].

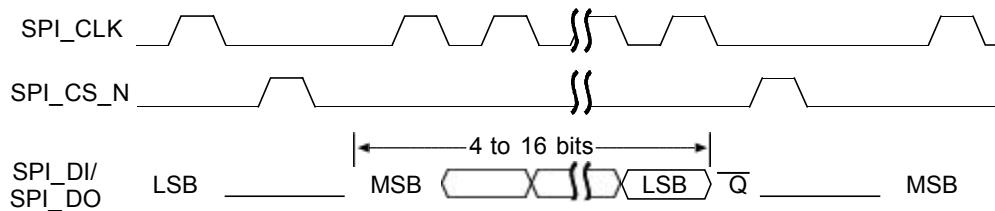
- SPO = 0, SPH = 0

[Figure 12-20](#) shows the SPI single frame format.

**Figure 12-20 SPI single frame format (SPO = 0, SPH = 0)**



[Figure 12-21](#) shows the SPI continuous frame format.

**Figure 12-21 SPI continuous frame format (SPO = 0, SPH = 0)**

When the SPI is idle in this mode:

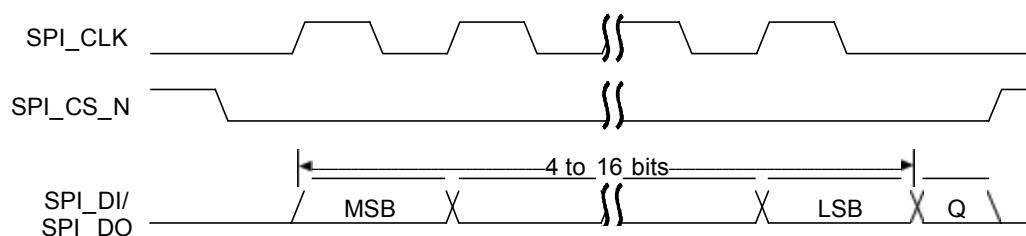
- The SPI\_CLK signal is set to low.
- The SPI\_CS\_N signal is set to high.
- The TX data line SPI\_DO is forced to low.

When the SPI is enabled and valid data is ready in the TX FIFO, data transfer starts if the SPI\_CS\_N signal is set to low. The data from the slave device is transferred to the RX data line SPI\_DI of the master device immediately. Half SPI\_CLK clock cycle later, the valid master data is transmitted to SPI\_DO. At this time, both the master and slave data become valid. The SPI\_CLK pin changes to high level in the next half SPI\_CLK clock cycle. Then, data is captured on the rising edge and is transmitted on the falling edge of the SPI\_CLK clock.

If a single word is transferred, SPI\_CS\_N is restored to high level one SPI\_CLK clock later after the last bit is captured.

For continuous transfer, the SPI\_CS\_N signal must pull up the SPI\_CLK clock by one clock cycle between the transfers of two words. When SPH is 0, the slave select pin fixes the data of the internal serial device register. Therefore, the master device must pull up the SPI\_CS\_N signal between the transfers of two words in continuous transfer. When the continuous transfer ends, SPI\_CS\_N is restored to high level one SPI\_CLK clock cycle later after the last bit is captured.

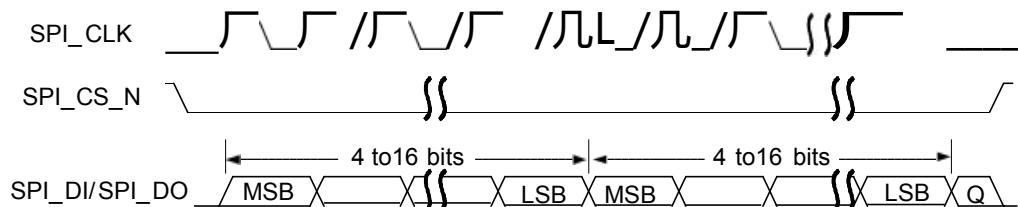
- SPO = 0, SPH = 1

**Figure 12-22 shows the SPI single frame format.****Figure 12-22 SPI single frame format (SPO = 0, SPH = 1)**



[Figure 12-23](#) shows the SPI continuous frame format.

**Figure 12-23 SPI continuous frame format (SPO = 0, SPH = 1)**



When the SPI is idle in this mode:

- The SPI\_CLK signal is set to low.
- The SPI\_CS\_N signal is set to high.
- The TX data line SPI\_DO is forced to low.

When the SPI is enabled and valid data is ready in the TX FIFO, data transfer start if the SPI\_CS\_N signal is set to low. The master and slave data become valid on their respective transfer line half SPI\_CLK clock cycle later. In addition, SPI\_CLK becomes valid from the first rising edge. Then, data is captured on the falling edge and is transmitted on the rising edge of the SPI\_CLK clock.

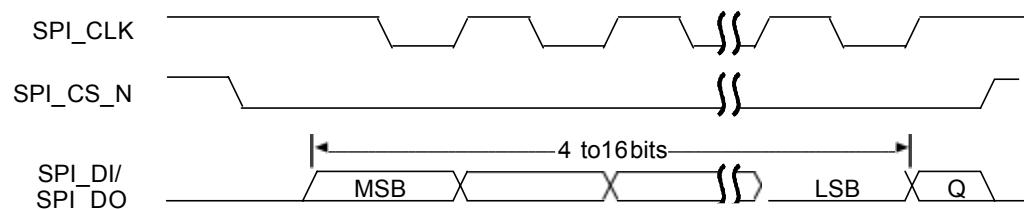
If a single word is transferred, SPI\_CS\_N is restored to high level one SPI\_CLK clock later after the last bit is captured.

For a continuous transfer, SPI\_CS\_N remains low between the transfers of two words. When the continuous transfer ends, SPI\_CS\_N is restored to high level one SPI\_CLK clock cycle later after the last bit is captured.

- SPO = 1, SPH = 0

[Figure 12-24](#) shows the SPI single frame format.

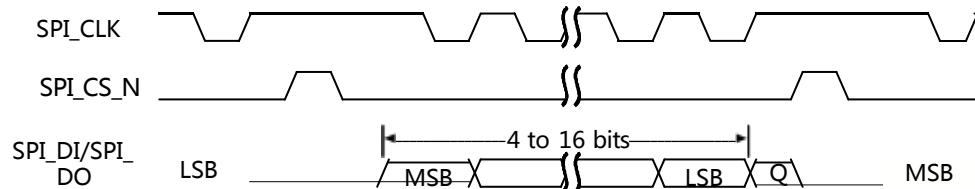
**Figure 12-24 SPI single frame format (SPO = 1, SPH = 0)**



[Figure 12-25](#) shows the SPI continuous frame format.



**Figure 12-25 SPI continuous frame format (SPO = 1, SPH = 0)**



When the SPI is idle in this mode:

- The SPI\_CLK signal is set to high.
- The SPI\_CS\_N signal is set to high.
- The TX data line SPI\_DO is forced to low.

When the SPI is enabled and valid data is ready in the TX FIFO, data transfer starts if the SPI\_CS\_N signal is set to low. The data from the slave device is transferred to the RX data line SPI\_DI of the master device immediately. Half SPI\_CLK clock cycle later, the valid master data is transmitted to SPI\_DO. After another half SPI\_CLK clock cycle, the SPI\_CLK master pin is set to low. Then, data is captured on the falling edge and is transmitted on the rising edge of the SPI\_CLK clock.

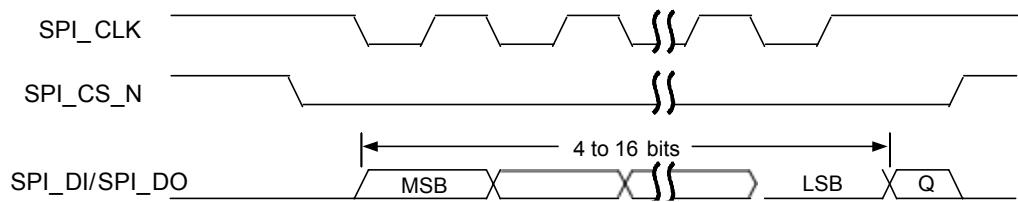
If a single word is transferred, SPI\_CS\_N is restored to high level after one SPI\_CLK clock since the data of the last bit is captured.

For a continuous transfer, the SPI\_CS\_N signal must be pulled up between the transfers of two words. This is because that when SPH is 0, the slave select pin fixes the data of the internal serial device register. SPI\_CS\_N is restored to high level one SPI\_CLK clock later after the last bit is captured.

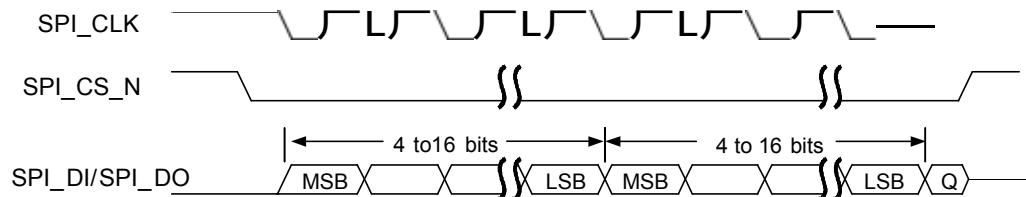
- SPO = 1, SPH = 1

[Figure 12-26](#) shows the SPI single frame format.

**Figure 12-26 SPI single frame format (SPO = 1, SPH = 1)**



[Figure 12-27](#) shows the SPI continuous frame format.

**Figure 12-27 SPI continuous frame format (SPO = 1, SPH = 1)**

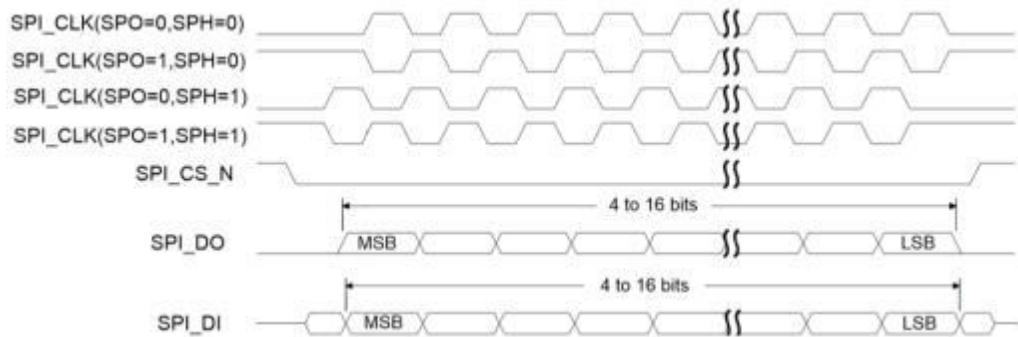
When the SPI is idle in this mode:

- The SPI\_CLK signal is set to high.
- The SPI\_CS\_N signal is set to high.
- The TX data line SPI\_DO is forced to low.

When the SPI is enabled and valid data is ready in the TX FIFO, data transfer starts if the SPI\_CS\_N signal is set to low. Half SPI\_CLK clock cycle later, the master data and slave data are valid on respective transfer line. In addition, SPI\_CLK becomes valid from a falling edge of SPI\_CLK. Then, data is captured on the rising edge and is transmitted on the falling edge of the SPI\_CLK clock. If a single word is being transmitted, SPI\_CS\_N is restored to high level one SPI\_CLK clock cycle later after the last bit is captured.

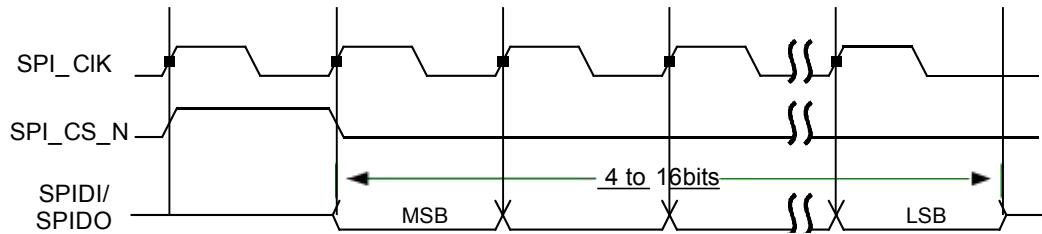
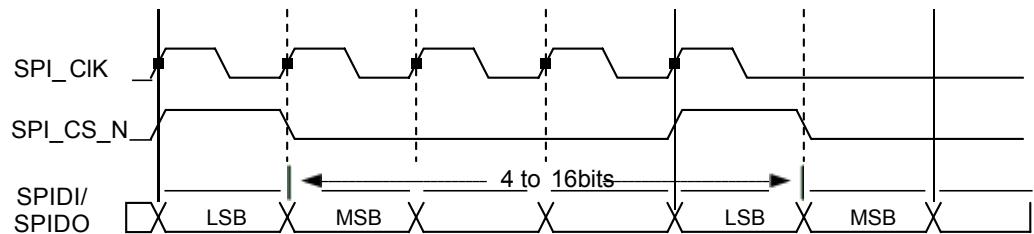
For a continuous transfer, the SPI\_CS\_N signal remains low. SPI\_CS\_N is restored to high level one SPI\_CLK clock cycle after the last bit is captured. For a continuous transfer, SPI\_CS\_N remains low during transfer. The method of ending data transfer is the same as that in single transfer mode.

- Interface timings

**Figure 12-28 SPI timings**

## TI Synchronous Serial Interface

Figure 12-29 shows the TI synchronous serial single frame format.

**Figure 12-29** TI synchronous serial single frame format**Figure 12-30** shows the TI synchronous serial continuous frame format.**Figure 12-30** TI synchronous serial continuous frame format

When the SPI is idle in this mode:

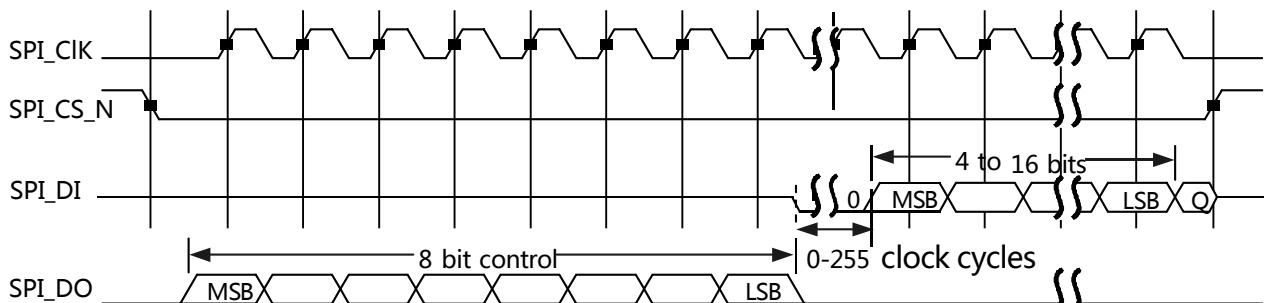
- The SPI\_CK signal is set to low.
- The SPI\_CS\_N signal is set to low.
- The transfer data line SPI\_DO retains high impedance.

If there is data in the TX FIFO, SPI\_CS\_N generates a high-level pulse in one SPI\_CK clock cycle. Then, the data to be transmitted is transferred from the TX FIFO to the TX logic serial shift register. In addition, the MSBs of the data frames with the length of 4–16 bits are shifted and output from SPI\_DO on the next rising edge of the SPI\_CK clock. Similarly, the MSB of the data received from the external serial slave device is shifted and input from the SPI\_DI pin.

The SPI and off-chip serial device stores the data in the serial shift register on the falling edge of the SPI\_CK clock. The RX serial register transmits the data to the RX FIFO on the rising edge of the first SPI\_CK clock after receiving the LSB.

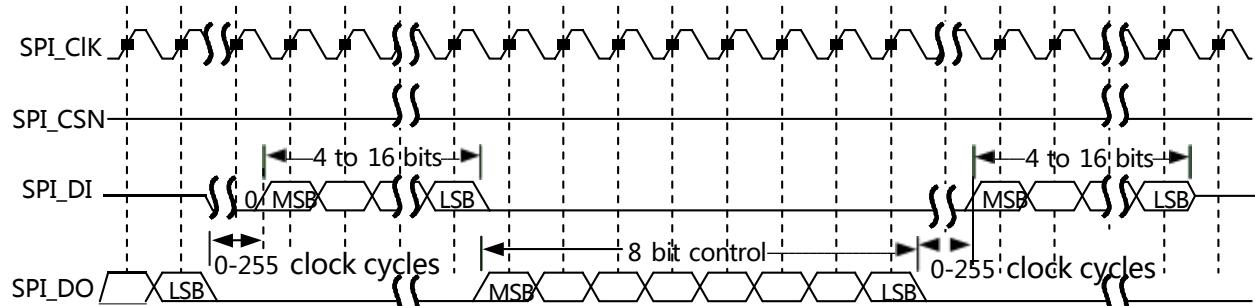
## National Semiconductor Microwire Interface

**Figure 12-31** shows the national semiconductor microwire single frame format.

**Figure 12-31** National semiconductor microwire single frame format

0 to 255 clock cycles can be delayed between the end of SPI\_DO LSB and the start of SPI\_DI MSB.

[Figure 12-32](#) shows the national semiconductor microwire continuous frame format.

**Figure 12-32** National semiconductor microwire continuous frame format

0 to 255 clock cycles can be delayed between the end of SPI\_DO LSB and the start of SPI\_DI MSB.

The microwire format is similar to the SPI format because both of them use the technology of transferring master-slave information. The only difference is that the SPI works in full-duplex mode and the microwire interface works in half-duplex mode. Before the SPI transmits serial data to the external chip, an 8-bit control word needs to be added. In this process, the SPI does not receive any data. After data transfer is complete, the external chip decodes the received data. One clock cycle later after 8-bit control information is transmitted, the slave device starts to respond to the required data. The returned data length is 4 bits to 16 bits, and the length of the entire frame is 13 bits to 25 bits.

When the SPI is idle in this mode:

- The SPI\_CLK signal is set to low.
- The SPI\_CS\_N signal is set to high level.
- The TX data signal SPI\_DO is forced to low level.

Writing one control byte to the TX FIFO starts a data transfer. The data transfer is triggered on a falling edge of SPI\_CS\_N. The data of the TX FIFO is sent to the serial shift register. The MSB of the 8-bit control frame is transmitted to the SPI\_DO pin.



During frame transfer, SPI\_CS\_N remains low. Whereas SPI\_DI retains high impedance.

The off-chip serial slave device latches the data to the serial shift register on each rising edge of the SPI\_CLK clock. After the slave device latches the data of the last bit, the slave device starts to decode the received data in the next clock cycle. Then, the slave device provides the data required by the SPI. Each bit is written to SPI\_DI on the falling edge of the SPI\_CLK clock. For a single data transfer, SPI\_CS\_N is pulled up at the end of the frame one clock cycle after the last bit is written to the RX serial register. In this way, the RX data is transmitted to the RX FIFO.

The start and end for a continuous data transfer are the same as those for a signal data transfer. During the continuous data transfer, SPI\_CS\_N retains low and the data transferred is continuous. The control word of the next frame is adjacent to the LSB of the previous frame. When the LSB of the frame is latched to the SPI, each received value is originated from the receive shift register on the falling edge of the SPI\_CLK clock.

## 12.4.5 Operating Mode

### Working Modes

The SPI working modes include the data transmission in the interrupt or query mode and the data transmission in the DMA mode.

### Clock and Reset

The frequency of the output SPI clock is calculated as follows:

$$F_{\text{sspclkout}} = F_{\text{sspclk}} / [\text{CPSDVSR} \times (1 + \text{SCR})]$$

**Fsspclk** is the working reference clock of the SPI and its value is 100 MHz.

For details about CPSDVSR and SCR, query [SPICPSR](#) and [SPICR0](#).

The SPI supports separate soft reset and clock gating. For details, see section 3.2.5 "CRG Register Description."

### Interrupts

The SPI has five interrupts. Four of them have separate interrupt sources and maskable and active high.

- **SPIRXINTR**  
RX FIFO interrupt request. When there are four or more valid data segments in the RX FIFO, the interrupt is set to 1.
- **SPITXINTR**  
TX FIFO interrupt request. When there are four or less valid data segments in the TX FIFO, the interrupt is set to 1.
- **SPIRORINTR**



RX overrun interrupt request. When the FIFO is full and new data is written to the FIFO, the FIFO is overrun and the interrupt is set to 1. In this case, the data is written to the RX shift register rather than the FIFO.

- SPIRTINTR

RX timeout interrupt request. When the RX FIFO is not empty and the SPI is idle for more than a fixed 32-bit cycle, the interrupt is set to 1.

It indicates that data in the RX FIFO needs to be transmitted. When the RX FIFO is empty or new data is transmitted to SPIRXD, the interrupt is cleared. The interrupt can also be cleared by writing to the SPIICR[RTIC] register.

- SPIINTR

Combined interrupt. This interrupt is obtained by performing an OR operation on the preceding four interrupts. If any of the preceding four interrupts is set to 1 and enabled, SPIINTR is set to 1.

For details about how to connect the SPIINTR of SPI, see "Interrupts" in this section.

## Initialization

The initialization is implemented as follows:

**Step 1** Write 0 to [SPICR1\[sse\]](#) to disable the SPI.

**Step 2** Write to [SPICR0](#) to set the parameters such as the frame format and transfer data bit width.

**Step 3** Configure [SPICPSR](#) to set the required clock divider.

**Step 4** In the interrupt mode, configure [SPIIMSC](#) to enable the corresponding interrupts or disable the generation of corresponding interrupts in query or DMA mode.

**Step 5** Configure [SPITXFIFOCR](#) and [SPIRXFIFOCR](#) in interrupt or DMA mode.

**Step 6** Configure [SPIDMACR](#) to enable the DMA function of the SPI in DMA mode.

----End

## Data Transfer in Query Mode

Perform the following steps:

**Step 1** Write 1 to [SPICR1\[sse\]](#) to enable the SPI.

**Step 2** Write the data to be transmitted to SPIDR continuously.

**Step 3** Poll [SPISR](#) until SPISR[BSY] is 0, SPISR[TFE] is 1, and SPISR[RNE] is 1. If SPISR[BSY] is 0, the bus is busy; if SPISR[TFE] is 1, the TX FIFO is empty; if SPISR[RNE] is 1, the RX FIFO is not empty.

**Step 4** Read data until the RX FIFO is empty. You can check whether the RX FIFO is empty by querying [SPISR\[RNE\]](#).



### NOTICE

The Motorola SPI and TI synchronous serial interface have full-duplex features. That is, a data segment is received after a data segment is transmitted. Even if only data is transmitted, the RX FIFO must be cleared.

**Step 5** Write 0 to [SPICR1\[sse\]](#) to disable the SPI.

----End

## Data Transfer in Interrupt Mode

Perform the following steps:

**Step 1** Write 1 to [SPICR1\[sse\]](#) to disable the SPI.

**Step 2** Write the data to be transmitted to SPIDR continuously.

**Step 3** Wait for the interrupt SPIRXINTR and read data in cyclic mode until all data is read.

Note that the full-duplex features of the SPI/Microwire. That is, a data segment is received after a data segment is transmitted. Even if only data is transmitted, the RX FIFO must be cleared.

### NOTICE

The Motorola SPI and TI synchronous serial interface has full-duplex features. That is, a data segment is received after a data segment is transmitted. Even if only data is transmitted, the RX FIFO must be cleared.

**Step 4** Write 0 to [SPICR1\[sse\]](#) to disable the SPI.

----End

## Data Transfer in DMA Mode

Perform the following steps:

**Step 1** Obtain a DMAC channel.

**Step 2** Write 1 to [SPICR1\[sse\]](#) to disable the SPI.

**Step 3** Transmit data.

- Configure the parameters of the configuration register and control register related to the DMAC channel.
- Start the DMAC and respond to the DMA request of the SPI TX FIFO for the data transfer.



- Check whether all data is transmitted by viewing the DMA interrupt. If all data is transmitted, disable the DMA transmit function of the SPI.

**Step 4** Receive data

- Configure the parameters of the configuration register and control register related to the DMAC channel.
- Start the DMAC and respond to the DMA request of the SSP RX FIFO for the data transfer.
- Check whether all data is received by viewing the DMA interrupt. If all data is received, disable the DMA receive function of the SPI.

**Step 5** Write 0 to [SPICR1\[sse\]](#) to disable the SPI.

----End

## 12.4.6 Register Summary

[Table 12-14](#)describes the registers.

- The base address of SPI0 registers is 0x1107\_0000.
- The base address of SPI1 registers is 0x1107\_1000.

**Table 12-14** Summary of SPI registers

Offset Address	Register	Description	Page
0x000	SPICR0	Control register 0	<a href="#">12-10 6</a>
0x004	SPICR1	Control register 1	<a href="#">12-10 7</a>
0x008	SPIDR	Data register	<a href="#">12-10 8</a>
0x00C	SPISR	Status register	<a href="#">12-10 9</a>
0x010	SPICPSR	Clock divider register	<a href="#">12-10 9</a>
0x014	SPIIMSC	Interrupt mask register	<a href="#">12-11 0</a>
0x018	SPIRIS	Raw interrupt status register	<a href="#">12-11 0</a>
0x01C	SPIMIS	Masked interrupt status register	<a href="#">12-11 1</a>



Offset Address	Register	Description	Page
0x020	SPIICR	Interrupt clear register	<a href="#">12-11 1</a>
0x024	SPIIDMACR	DMA control register	<a href="#">12-11 1</a>
0x028	SPITXFIFOOCR	TX FIFO control register	<a href="#">12-11 2</a>
0x02C	SPIRXFIFOOCR	RX FIFO control register	<a href="#">12-11 3</a>

## 12.4.7 Register Description

### SPICR0

SPICR0 is SPI control register 0.

Offset Address: 0x000 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	RW	SCR	Serial clock rate, ranging from 0 to 255. The value of the SCR is used to generate the TX and RX bit rates of the SPI. The formula is as follows: $F_{sspclk}/(CPSDVSR \times (1 + SCR))$ . CPSDVSR is an even ranging from 2 to 254, and it is configured by SPICPSR.	0x00
[7]	RW	SPH	SPICLKOUT phase. For details, see section <a href="#">12.4.4 "Peripheral Bus Timings."</a>	0x0
[6]	RW	SPO	SPICLKOUT polarity. For details, see section <a href="#">12.4.4 "Peripheral Bus Timings."</a>	0x0
[5:4]	RW	FRF	Frame format select. 00: Motorola SPI frame format 01: TI synchronous serial frame format 10: National microwire frame format 11: reserved	0x0
[3:0]	RW	DSS	Data width 0011: 4 bits 0100: 5 bits	0x0



Bits	Access	Name	Description	Reset
			0101: 6 bits 0110: 7 bits 0111: 8 bits 1000: 9 bits 1001: 10 bits 1010: 11 bits 1011: 12 bits 1100: 13 bits 1101: 14 bits 1110: 15 bits 1111: 16 bits Other values: reserved	

## SPICR1

SPICR1 is SPI control register 1.

Offset Address: 0x004 Total Reset Value: 0x7F00

Bits	Access	Name	Description	Reset
[15]	RW	WaitEn	Wait enable. This bit is valid when the SPICR0[FRF] is set to the national microwire frame format. 0: disabled 1: enabled	0x0
[14:8]	RW	WaitVal	Number of waiting beats between read and write when in the national microwire frame format. When WaitEn is 1 and the frame format is national microwire, WaitVal is valid.	0x7F
[7]	-	reserved	Reserved	0x0
[6]	RW	mode_altasens	0: The CS signal is automatically generated by the SoC logic based on the selected timing. 1: The CS signal is controlled by the SPI enable when the Motorola SPI frame format is used. If the SPI is enabled, the CS signal is pulled down; otherwise, the	0x0



Bits	Access	Name	Description	Reset
			CS signal is pulled up.	
[5]	-	reserved	Reserved	0x0
[4]	RW	BitEnd	Data order control 0: MSB to LSB 1: LSB to MSB	0x0
[3]	-	reserved	Reserved	0x0
[2]	RW	MS	Master or slave mode. This bit can be changed only when the SPI is disabled. 0: master mode 1: slave mode	0x0
[1]	RW	SSE	SPI enable 0: disabled 1: enabled	0x0
[0]	RW	LBM	Loopback mode 0: A normal serial port operation is enabled. 1: The output of the TX serial shift register is connected to the input of the RX serial shift register.	0x0

## SPIDR

SPIDR is a data register.

Offset Address: 0x008 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	DATA	TX or RX FIFO Read: RX FIFO Write: TX FIFO If the data is less than 16 bits, the data must be aligned to the right. The TX logic ignores the unused upper bits, and the RX logic automatically aligns the data to the right.	0x0000



## SPISR

SPISR is a status register.

Offset Address: 0x00C Total Reset Value: 0x0003

Bits	Access	Name	Description	Reset
[15:5]	-	reserved	Reserved	0x000
[4]	RO	BSY	SPI busy flag 0: idle 1: busy	0x0
[3]	RO	RFF	Whether the RX FIFO is full 0: not full 1: full	0x0
[2]	RO	RNE	Whether the RX FIFO is not empty 0: empty 1: not empty	0x0
[1]	RO	TNF	Whether the TX FIFO is not full 0: full 1: not full	0x1
[0]	RO	TFE	Whether the TX FIFO is empty 0: not empty 1: empty	0x1

## SPICPSR

SPICPSR is a clock divider register.

Offset Address: 0x010 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x00
[7:0]	RW	CPSDVSR	Clock divider. The value must be an even ranging from 2 to 254. It depends on the frequency of the input clock SPICLK. The LSB is read as 0.	0x00



## SPIIMSC

SCIIMSC is an interrupt mask register. The value 0 indicates an interrupt is masked and the value1 indicates an interrupt is not masked.

Offset Address: 0x014 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x000
[3]	RW	TXIM	TX FIFO interrupt mask 0: The interrupt is masked when only half of or less data is left in the TX FIFO. 1: The interrupt is not masked when only half of or less data is left in the TX FIFO.	0x0
[2]	RW	RXIM	RX FIFO interrupt mask 0: The interrupt is masked when only half of or less data is left in the RX FIFO. 1: The interrupt is not masked when only half of or less data is left in the RX FIFO.	0x0
[1]	RW	RTIM	RX timeout interrupt mask 0: masked 1: not masked	0x0
[0]	RW	RORIM	RX overflow interrupt mask 0: masked 1: not masked When the value is 1, the hardware stream control function is enabled. That is, when the RX FIFO is full, the SPI stops transmitting data.	0x0

## SPIRIS

SPIRIS is a raw interrupt status register. The value 0 indicates no interrupts are generated, and the value 1 indicates interrupts are generated.

Offset Address: 0x018 Total Reset Value: 0x0008

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x000
[3]	RO	TXRIS	Raw TX FIFO interrupt status	0x1



[2]	RO	RXRIS	Raw RX FIFO interrupt status	0x0
[1]	RO	RTRIS	Raw RX timeout interrupt status	0x0
[0]	RO	RORRIS	Raw RX overflow interrupt status	0x0

## SPIMIS

SPIMIS is a masked interrupt status register. The value 0 indicates no interrupts are generated, and the value 1 indicates interrupts are generated.

Offset Address: 0x01C Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x000
[3]	RO	TXMIS	Status of the masked TX FIFO interrupt	0x0
[2]	RO	RXMIS	Status of the masked RX FIFO interrupt	0x0
[1]	RO	RTMIS	Status of the masked RX timeout interrupt	0x0
[0]	RO	RORMIS	Status of the masked RX overflow interrupt	0x0

## SPIICR

SCIICR is an interrupt clear register. Writing 1 clears an interrupt, and writing 0 has no effect.

Offset Address: 0x020 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:2]	-	reserved	Reserved	0x0000
[1]	WC	RTIC	RX timeout interrupt clear	0x0
[0]	WC	RORIC	RX overflow interrupt clear	0x0

## SPIDMACR

SCIDMACR is a DMA control register.

Offset Address: 0x024 Total Reset Value: 0x0000



Bits	Access	Name	Description	Reset
[15:2]	-	reserved	Reserved	0x0000
[1]	RW	TXDMAE	DMA TX FIFO enable 0: disabled 1: enabled	0x0
[0]	RW	RXDMAE	DMA RX FIFO enable 0: disabled 1: enabled	0x0

## SPITXFIFOCCR

SPITXFIFOCCR is a TX FIFO control register.

Offset Address: 0x028 Total Reset Value: 0x0009

Bits	Access	Name	Description	Reset
[15:6]	-	reserved	Reserved	0x000
[5:3]	RW	TXINTSize	Threshold for generating the TX FIFO request interrupt. That is, when the number of data segments in the TX FIFO is less than or equal to the value of TXINTSize, TXRIS is valid. 000: 1 001: 4 010: 8 011: 16 100: 32 101: 64 110: 128 111: 128	0x1
[2:0]	RW	DMATXBRSIZE	Threshold for generating the TX FIFO request DMA transfer burst. That is, when the number of data segments in the TX FIFO is less than or equal to the configured value (256 - DMATXBRSIZE), DMATXBREQ is valid. The width of the TX FIFO is 16 bits. 000: 1	0x1



Bits	Access	Name	Description	Reset
			001: 4 010: 8 011: 16 100: 32 101: 64 110: 64 111: 64	

## SPIRXFIFOCR

SPIRXFIFOCR is an RX FIFO control register.

Offset Address: 0x02C Total Reset Value: 0x0009

Bits	Access	Name	Description	Reset
[15:6]	-	reserved	Reserved	0x000
[5:3]	RW	RXINTSize	Threshold for generating the RX FIFO request interrupt. That is, when the number of data segments in the TX FIFO is less than or equal to the configured value (256 - RXINTSize), RXRIS is valid. The width of the RX FIFO is 16 bits. 000: 1 001: 4 010: 8 011: 16 100: 32 101: 64 110: 128 111: 192	0x1
[2:0]	RW	DMARXBRSIZE	Burst transfer threshold. When this threshold is reached, the RX FIFO asks the DMA to perform a burst transfer. That is, when number of data segments in the TX FIFO is less than or equal to the value of DMARXBRSIZE, DMARBREQ is valid. 000: 1 001: 4	0x1



Bits	Access	Name	Description	Reset
			010: 8 011: 16 100: 32 101: 64 110: 96 111: 128	

## 12.5 GPIO

### 12.5.1 Overview

The SoC supports 11 GPIO groups, namely GPIO0–GPIO10. Each group provides eight programmable I/O 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. As input, each GPIO pin can act as an interrupt source; as output, each GPIO pin can be set to 0 or 1 separately.

The GPIO can generate maskable interrupts based on the level or transition value. The general purpose input output interrupt (GPIOINTR) signal provides an indicator to the interrupt controller, indicating that an interrupt occurs.

#### NOTICE

For details about the number of GPIO pins and multiplexing relationship between GPIO pins and other pins and control modes, see *Hi35xxVxxx\_PINOUT\_EN.xlsx*. (Hi35xxVxxx indicates an SoC version.)

For the multiplexed GPIO pins that are output by default, note that the pins that connect to the SoC and the components must be input.

### 12.5.2 Features

Each GPIO pin can be configured as input or output.

- As input, each GPIO pin can act as an interrupt source.
- As output, each GPIO can be set to 0 or 1 separately.



## 12.5.3 Operating Mode

### Interface Reset

The GPIO is reset during SoC power-on reset or system reset, and GPIO pins are in input state after being reset.

### GPIO

Each pin can be configured as input or output. To configure a GPIO pin, perform the following steps:

- Step 1** Enable the required GPIO pins by configuring corresponding pins according to the description of multiplexing registers.
- Step 2** Configure the GPIO as input or output by using the[GPIO\\_DIR](#)register.
- Step 3** When the GPIO pins are in input mode, read the[GPIO\\_DATA](#)register to check the input signal value. When the GPIO pins are in output mode, write the output value to the[GPIO\\_DATA](#)register to control the output level of the GPIO pins.  
----End

#### NOTICE

When the GPIO pins are in output mode, do not enable the GPIO interrupt. Otherwise, the GPIO interrupt will be generated when output signals meet interrupt generation conditions.

### Interrupt Operation

The GPIO interrupt is controlled through seven registers (such as[GPIO\\_IS](#)). These registers enable you to select the interrupt source, interrupt edge (falling edge or rising edge), and interrupt trigger modes (level-sensitive mode or edge-sensitive mode). For details about the corresponding interrupt registers of the GPIO, see section 3.3 "Interrupt Systems."

When multiple interrupts are generated at the same time, these interrupts are combined into an interrupt and then reported. For details about the GPIO interrupt mapping, see section 3.3 "Interrupt Systems."

The[GPIO\\_IS](#),[GPIO\\_IBE](#), and[GPIO\\_IEN](#)registers determine the features of the interrupt source and interrupt trigger type.

[GPIO\\_RIS](#)and[GPIO\\_MIS](#)are used to read the raw interrupt status and masked interrupt status, respectively. [GPIO\\_IEN](#) controls the final report status of each interrupt. In addition,[GPIO\\_IC](#)is provided to clear the interrupt status.

Perform the following operations to configure GPIO pins to the interrupt mode:



- Step 1** Select the edge-sensitive mode or level-sensitive mode by configuring the[GPIO\\_IS](#) register.
- Step 2** Select the falling-/rising-edge-sensitive mode or high-/low-level-sensitive mode by configuring the[GPIO\\_IEV](#) register.
- Step 3** If the edge-sensitive mode is selected, select single-edge-sensitive mode or dual-edge-sensitive mode by configuring the[GPIO\\_IBE](#) register.
- Step 4** Write 0xFF to the[GPIO\\_IC](#) register to clear the interrupt.
- Step 5** Set the[GPIO\\_IE](#) to 1 to enable the interrupt.

----End

### NOTICE

Ensure that data in GPIO pins is stable during initialization to prevent pseudo interrupts.

The GPIO interrupts are controlled by seven registers. When one or more GPIO pins generate interrupts, a combined interrupt is output to the interrupt controller. The differences between the edge-sensitive mode and level-sensitive mode are as follows:

- Edge-sensitive mode: Software must clear this interrupt to enable superior interrupts.
- Level-sensitive mode: The external interrupt source must keep this level until the processor identifies this interrupt.

## 12.5.4 Register Summary

[Table 12-15](#)describes the base addresses for the GPIO registers.

**Table 12-15** Base addresses for GPIO registers

GPIO Controller	Base Address
GPIO10	0x1109_A000
GPIO9	0x1109_9000
GPIO8	0x1109_8000
GPIO7	0x1109_7000
GPIO6	0x1109_6000
GPIO5	0x1109_5000



GPIO Controller	Base Address
GPIO4	0x1109_4000
GPIO3	0x1109_3000
GPIO2	0x1109_2000
GPIO1	0x1109_1000
GPIO0	0x1109_0000

### NOTE

Register address of GPIOn = GPIOn base address + Offset address of the register

**Table 12-16**describes the offset addresses and definitions of a group of internal GPIO registers. GPIO0 to GPIOn also have the same internal GPIO registers.

**Table 12-16** Summary of GPIO registers

Offset Address	Register	Description	Page
0x000-0x3FC	GPIO_DATA	GPIO data register	<a href="#">12-117</a>
0x400	GPIO_DIR	GPIO direction control register	<a href="#">12-118</a>
0x404	GPIO_IS	GPIO interrupt trigger register	<a href="#">12-119</a>
0x408	GPIO_IBE	GPIO interrupt dual-edge trigger register	<a href="#">12-119</a>
0x40C	GPIO_IEV	GPIO interrupt trigger event register	<a href="#">12-119</a>
0x410	GPIO_IE	GPIO interrupt mask register	<a href="#">12-120</a>
0x414	GPIO_RIS	GPIO raw interrupt status register	<a href="#">12-120</a>
0x418	GPIO_MIS	GPIO masked interrupt status register	<a href="#">12-121</a>
0x41C	GPIO_IC	GPIO interrupt clear register	<a href="#">12-121</a>

## 12.5.5 Register Description

### GPIO\_DATA

GPIO\_DATA is a GPIO data register. It is used to buffer the input or output data.



When the corresponding bit of the [GPIO\\_DIR](#) is configured as output, the values written to the [GPIO\\_DATA](#) register are sent to the corresponding pin (note that the pin multiplexing configuration must be correct). If the bit is configured as input, the value of the corresponding input pin is read.

### NOTICE

If the corresponding bit of [GPIO\\_DIR](#) is configured as input, the pin value is returned after a valid read; if the corresponding bit is configured as output, the written value is returned after a valid read.

Through PADDR[9: 2], the [GPIO\\_DATA](#) register masks the read and write operations on the register. The register corresponds to 256 address spaces. PADDR[9: 2] corresponds to [GPIO\\_DATA](#)[7: 0]. When the corresponding bit is high, it can be read or written. When the corresponding bit is low, no operations are supported. For example:

- ⌚ If the address is 0x3FC (0b11\_1111\_1100), the operations on all the eight bits of [GPIO\\_DATA](#) bit[7: 0] are valid.
- ⌚ If the address is 0x200 (0b10\_0000\_0000), only the operation on [GPIO\\_DATA](#) bit[7] is valid.

Offset Address: 0x000-0x3FC Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	gpio_data	Indicates the GPIO input data when the GPIO is configured as input; indicates the GPIO output data when the GPIO is configured as output. Each bit can be controlled separately. The register is used together with <a href="#">GPIO_DIR</a> .	0x00

## GPIO\_DIR

[GPIO\\_DIR](#) is a GPIO direction control register. It is used to configure the direction of each GPIO pin.

Offset Address: 400 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	gpio_dir	GPIO direction control register. Bit[7: 0] correspond to <a href="#">GPIO_DATA</a> [7: 0] respectively. Each bit can be controlled separately.	0x00



Bits	Access	Name	Description	Reset
			0: input 1: output	

## GPIO\_IS

GPIO\_IS is a GPIO interrupt trigger register. It is used to configure the interrupt trigger mode.

Offset Address: 404 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	gpio_is	GPIO interrupt trigger control register. Bit[7: 0] correspond to <a href="#">GPIO_DATA</a> [7: 0]. Each bit is controlled separately. 0: edge-sensitive mode 1: level-sensitive mode	0x00

## GPIO\_IBE

GPIO\_IBE is a GPIO interrupt dual-edge trigger register. It is used to configure the edge trigger mode of each GPIO pin.

Offset Address: 408 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	gpio_ibc	GPIO interrupt edge control register. Bit[7: 0] correspond to <a href="#">GPIO_DATA</a> [7: 0] respectively. Each bit is controlled independently. 0: single-edge-sensitive mode. The <a href="#">GPIO_IEV</a> register controls whether the interrupt is rising-edge-sensitive or falling-edge-sensitive. 1: dual-edge-sensitive mode	0x00

## GPIO\_IEV

GPIO\_IEV is a GPIO interrupt event register. It is used to configure the interrupt trigger event of each GPIO pin.



Offset Address: 40C Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	gpio_iev	GPIO interrupt trigger event register. Bit[7: 0] correspond to <a href="#">GPIO_DATA</a> [7: 0]. Each bit is controlled separately. 0: falling-edge-sensitive mode or low-level-sensitive mode 1: rising-edge-sensitive mode or high-level-sensitive mode.	0x00

## GPIO\_IE

GPIO\_IE is a GPIO interrupt mask register. It is used to mask GPIO interrupts.

Offset Address: 410 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RW	gpio_ie	GPIO interrupt mask register. Bit[7: 0] correspond to <a href="#">GPIO_DATA</a> [7: 0]. Each bit is controlled separately. 0: masked 1: not masked	0x00

## GPIO\_RIS

GPIO\_RIS is a GPIO raw interrupt status register. It is used to query the raw interrupt status of each GPIO pin.

Offset Address: 414 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RO	gpio_ris	GPIO raw interrupt status register. Bit[7: 0] correspond to <a href="#">GPIO_DATA</a> [7: 0], indicating the unmasked interrupt status. The status cannot be masked and controlled by the <a href="#">GPIO_IE</a> register. 0: No interrupt occurs. 1: An interrupt is generated.	0x00



## GPIO\_MIS

GPIO\_MIS is a GPIO masked interrupt status register. It is used to query the masked interrupt status of each GPIO pin.

Offset Address: 418 Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	RO	gpio_mis	GPIO masked interrupt status register. Bit[7: 0] correspond to <a href="#">GPIO_DATA</a> [7: 0], indicating the masked interrupt status. The status is controlled by the <a href="#">GPIO_IE</a> register. 0: The interrupt is invalid. 1: The interrupt is valid.	0x00

## GPIO\_IC

GPIO\_IC is a GPIO interrupt clear register. It is used to clear the interrupts generated by GPIO pins and clear the [GPIO\\_RIS](#) and [GPIO\\_MIS](#) registers.

Offset Address: 41C Total Reset Value: 0x00

Bits	Access	Name	Description	Reset
[7:0]	WC	gpio_ic	GPIO interrupt clear register. Bit[7: 0] correspond to <a href="#">GPIO_DATA</a> [7: 0]. Each bit is controlled separately. 0: no effect 1: cleared	0x00

## 12.6 USB DRD

### 12.6.1 Overview

The chip supports one USB 2.0 dual role device (DRD) interface and can work in USB 2.0 host or USB 2.0 device mode, which cannot be dynamically switched.

- The USB 2.0 host supports three transfer rates defined in the USB 2.0 protocol—480 Mbit/s, 12 Mbit/s, and 1.5 Mbit/s, and four basic transfer modes—control transfer, bulk transfer, isochronous transfer, and interrupt transfer. It also integrates one Root Hub, supports a maximum of 64 devices at the same time, and supports the XHCI protocol.

- The USB 2.0 device supports two transfer rates—480 Mbit/s and 12 Mbit/s, and four basic transfer modes—control transfer, bulk transfer, isochronous transfer, and interrupt transfer. It also supports three IN ports and three OUT ports (except the control endpoint).

The USB DRD performs the following functions:

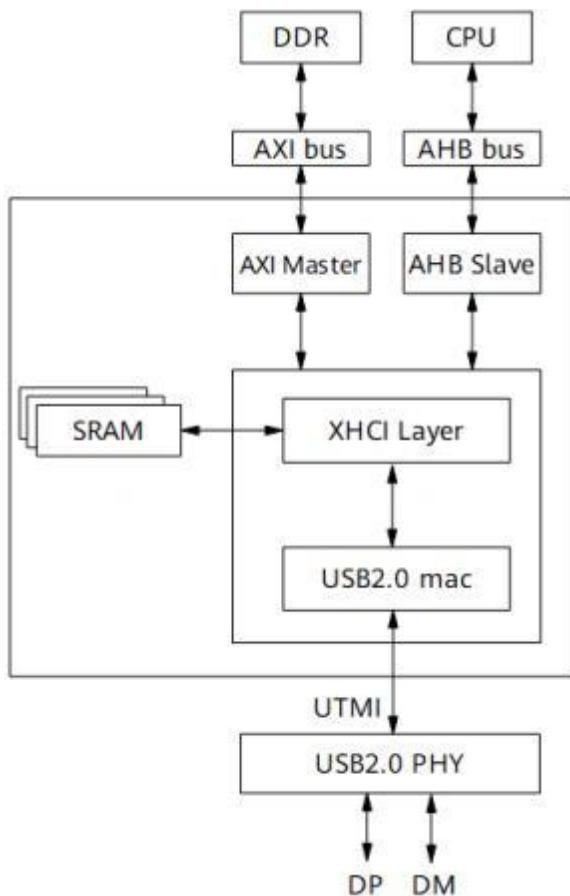
- Control and process protocols.
- Parse and package data.
- Encode and decode the transmitted signals.
- Provide the interrupt vector interface for the driver.

## 12.6.2 Function Description

### Logical Block Diagram

[Figure 12-33](#) shows the logical block diagram of the USB 2.0 DRD module.

**Figure 12-33** Logical block diagram of the USB 2.0 DRD module





### NOTE

- UTMI: USB2.0 transceiver macrocell interface
- XHCI: eXtensible host controller interface

## Features

The USB 2.0 DRD module has the following features:

- Supports the USB 2.0 protocol and is backward compatible with the USB 1.1 protocol.
- Supports the host mode or device mode, which cannot be dynamically switched.
- The host mode supports three transfer rates—480 Mbit/s, 12 Mbit/s, and 1.5 Mbit/s.
- The device mode supports two transfer rates—480 Mbit/s and 12 Mbit/s, and supports three IN ports and two OUT ports (except the control endpoint).
- Supports the USB 2.0 suspend, resume, and remote wakeup low-power features.
- Supports four basic transfer modes—control transfer, bulk transfer, isochronous transfer, and interrupt transfer.
- Supports direct memory access (DMA).
- The host mode supports an external USB hub to connect to maximum of 64 devices.

## Function Implementation

The USB 2.0 module supports the following transfer types in host mode:

- Control transfer  
This mode applies to the data transfer between endpoints 0 of the USB host and USB device. The control transfer is bidirectional and the transferred data amount is small. Depending on the device and transfer speed, 8-byte data, 16-byte data, 32-byte data, or 64-byte data can be transferred.
- Bulk transfer  
This mode applies to the data transfer in bulk when there is no limit on the bandwidth and time interval. This mode is the best choice when the transfer speed is very low and many data transfers are delayed. Bulk transfer is performed after all other types of data transfers are complete. In bulk transfer mode, data is transferred between the USB host and USB device without errors by using an error detection and retransmission mechanism.
- Isochronous transfer  
This mode is used for stream data transfer with strict time requirements and high error tolerance. It can also be used for instant data transfer at a constant transfer rate. This mode provides specific bandwidth and time interval.
- Interrupt transfer



This mode applies to transfer of small-sized, scattered, and unpredictable data. In this mode, the device is regularly checked for interrupt data to be sent. The query frequency ranges from 1 ms to 255 ms, depending on the device endpoint mode. Typically, the interrupt transfer is unidirectional and only input is available for the USB host.

### 12.6.3 Operating Mode

#### Clock and Reset of USB 2.0 DRD

The clock and reset registers must be configured before the USB controller is initialized.

Perform the following steps:

- Step 1** Write **0x1** to PERI\_CRG3632[4] to enable the reference clock of USB2.0 PHY.
- Step 2** Write **0x0** to PERI\_CRG3632[0] to deassert the power-on reset on the USB 2.0 PHY, with the delay of 1 ms.
- Step 3** Write **0x3** to USB2\_PHY\_ANA\_CFG5[1:0] to keep the PLL always on when the USB 2.0 PHY is in standby mode.
- Step 4** Write **0x0** to PERI\_CRG3632[1] to deassert the digital reset on the USB 2.0 PHY.
- Step 5** Write **0x0** to PERI\_CRG3634[0] to deassert the reset on the USB 2.0 controller.

----End

### 12.6.4 Eye Pattern Parameters of USB 2.0 PHY

The USB DRD controller corresponds to the USB 2.0 PHY.

#### NOTE

During PHY parameter adjustment, note that only values of the corresponding control bits need to be adjusted. Remain values of other bits unchanged.

### 12.6.5 Register Summary

[Table 12-17](#)describes the USB 2.0 PHY registers.

**Table 12-17** Summary of USB PHY registers (base address: 0x1031\_0000)

Offset Address	Register	Description	Page
0x0000	USB2_PHY_ANA_CF_G0	USB 2.0 PHY pre-drive adjustment register	<a href="#">12-125</a>
0x0010	USB2_PHY_ANA_CF_G4	USB 2.0 PHY high-speed eye pattern amplitude configuration register	<a href="#">12-125</a>



Offset Address	Register	Description	Page
0x0014	USB2_PHY_ANA_CF G5	USB 2.0 PHY PLL configuration register	<a href="#">12-125</a>

## 12.6.6 Register Description

### USB2\_PHY\_ANA\_CFG0

USB2\_PHY\_ANA\_CFG0 is the USB 2.0 PHY pre-drive adjustment register.

Offset Address: 0x0000 Total Reset Value: 0x0A33\_C003

Bits	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27:24]	RW	usb2_phy_rg_hstx_pdrv_p	Pre-drive strength level: bit<0>*1+bit<1>*2+bit<2>*4+bit<3> *8	0xA
[23:0]	-	reserved	Reserved	0x33C00 3

### USB2\_PHY\_ANA\_CFG4

USB2\_PHY\_ANA\_CFG4 is the USB 2.0 PHY high-speed eye pattern amplitude configuration register.

Offset Address: 0x0010 Total Reset Value: 0x0000\_0643

Bits	Access	Name	Description	Reset
[31:7]	-	reserved	Reserved	0x000000C
[6:4]	RW	usb2_phy_rg_vtxref_sel	Reference value of the high-speed eye pattern amplitude (representing only the adjustment trend instead of the accurate value)  000: 380 mV 001: 390 mV 010: 400 mV 011: 410 mV 100: 420 mV 101: 430 mV	0x4



Bits	Access	Name	Description	Reset
			110: 440 mV 111: 460 mV	
[3:0]	-	reserved	Reserved	0x3

## USB2\_PHY\_ANA\_CFG5

USB2\_PHY\_ANA\_CFG5 is the USB 2.0 PHY PLL configuration register.

Offset Address: 0x0014    Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	rg_pll_en_cfg	PLL enable. When this bit is set to 1, the PLL enable signal is controlled by <b>rg_pll_en_value</b> and is not affected by the <b>suspend</b> command, active high. 0: invalid 1: valid	0x0
[0]	RW	rg_pll_en_val	Configured value of PLL enable 0: disabled 1: enabled	0x0

## 12.7 LSADC

### 12.7.1 Overview

The low speed ADC (LSADC) measures analog signals by converting external analog signals into values in proportion, which can be applied in battery level detection and key detection.

The chip provides one LSADC.

- Hi3516CV610-10B/20B/20S/20G supports two independent channels.
- Hi3516CV610-00B/00S/00G supports four independent channels.



## 12.7.2 Features

The LSADC has the following features:

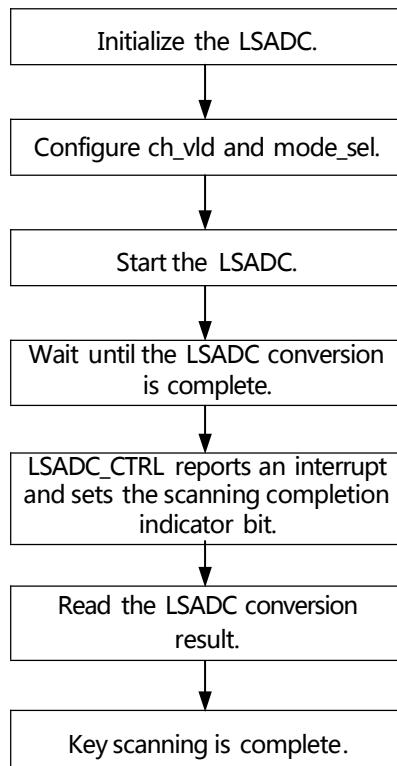
- Supports 3.3 V power voltage.
- Supports maximum scanning frequency of 200,000 times per second.
- Supports 10-bit sampling.
- Supports single scanning mode and continuous scanning mode.
- Supports 16-times and 32-times average algorithm modes.
- Supports automatic interrupt reporting when scanning is complete.
- Supports multi-channel scanning to prevent crosstalk.

## 12.7.3 Operating Mode

### Single Scanning Procedure

In single read mode (`LSADC_CTRL0[model_sel]` = 0), the CPU configures the ID of the channel to be scanned (only one channel can be configured), scanning mode, and key value mapping table and starts the LSADC to complete a channel scanning. After channel scanning is complete and the system is informed of the completion by using an interrupt, the CPU obtains the conversion result.

**Figure 12-34** Single scanning procedure

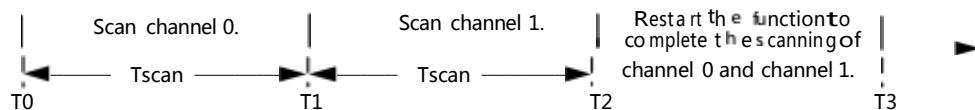


## Continuous Scanning Procedure

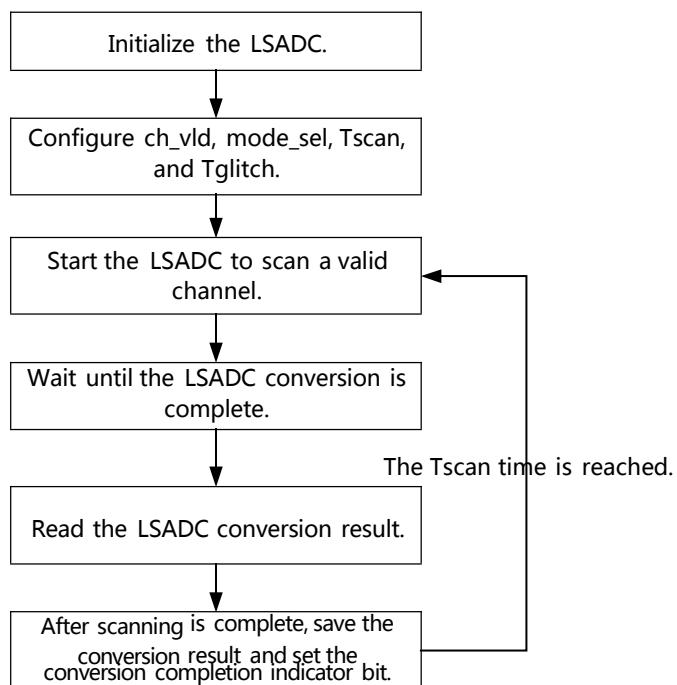
In continuous scanning mode (`LSADC_CTRL0[model_sel]` = 1), the CPU configures the continuous scanning interval (`Tscan`), glitch width (`Tglitch`), and valid channel ID (`ch_vld`) based on the application scenario, and starts the LSADC. The LSADC scans a valid channel (set the `LSADC_CTRL0[channel validity flag bit]` to valid) within `Tscan`, and scans the next valid channel when the next scanning time reaches. After all valid channels are scanned, the LSADC cyclically scans them again, as shown in [Figure 12-36](#).

[Figure 12-35](#) illustrates channel polling in continuous scanning mode.

**Figure 12-35** Take enabling channel 0 and channel 1 as an example, channel polling in continuous scanning mode



**Figure 12-36** Continuous scanning procedure





## Deglitch Procedure

The deglitch circuit adopts the majority decision algorithm. In the deglitch window, if the levels sampled by the ADC are the levels when keys are pressed, the value is considered as a valid key value; otherwise, a glitch signal is considered.

- No deglitch operation is performed in single read mode.
- The deglitch window Tglitch needs to be configured properly to enable the deglitch function in continuous scanning mode. For details, see the [LSADC\\_CTRL1](#) register.

## Sampling Depth Configuration

Use [LSADC\\_CTRL9](#) bit[9: 0] to set the sampling depth. The sampling depth can be set as required.

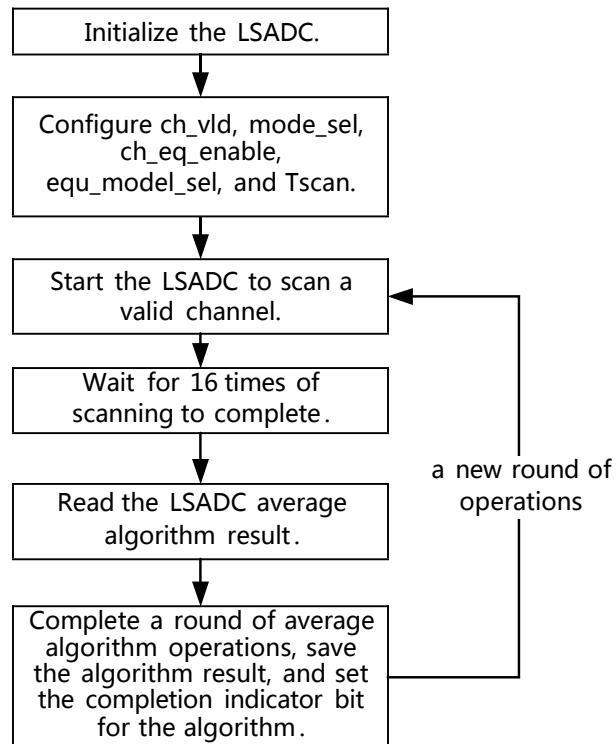
- When sampling depth is set to 10 bits, all the 10 bits of the sampling result are valid.
- When the sampling depth is less than 10 bits, only the upper bits of the sampling result are valid. For example, when the sampling depth is set to 8 bits, only the upper 8 bits of the sampling result are valid.

## Average Algorithm Configuration

The average algorithm mode can be selected for channel sampling by using [LSADC\\_CTRL0](#) bit[12]. You can set the average algorithm mode based on application requirements. For details, see the description of [LSADC\\_CTRL0](#).

- In single read mode, the average algorithm for sampling is not supported.
- In continuous scanning mode, the 16- or 32-time average algorithm can be used.

Take the 16-time average algorithm as an example. [Figure 12-37](#) shows the average algorithm of the channel in continuous scanning mode.

**Figure 12-37** Process of the 16-time average algorithm

## Configuration of the Anti-crosstalk Mode

Based on the continuous scanning configuration, you can prevent crosstalk between channels by configuring the following registers:

- Anti-crosstalk mode enable: Set [LSADC\\_CTRL0](#)bit[24] and [LSADC\\_CTRL0](#)bit[16] to 1.
- Anti-crosstalk period: Set [LSADC\\_CTRL0](#)bit[28:25] and [LSADC\\_CTRL0](#)bit[7:4]. (Note: The value of [LSADC\\_CTRL0](#)bit[7:4] must be greater than or equal to that of [LSADC\\_CTRL0](#)bit[28:25].)

### 12.7.4 Register Summary

[Table 12-18](#)describes LSADC registers.

**Table 12-18** Summary of LSADC registers (base address: 0x1110\_0000)

Offset Address	Register	Description	Page
0x0000	LSADC_CTRL0	LSADC configuration register	<a href="#">12-131</a>
0x0004	LSADC_CTRL1	Deglitch configuration register	<a href="#">12-134</a>



Offset Address	Register	Description	Page
0x0008	LSADC_CTRL2	Scanning interval configuration register	<a href="#">12-135</a>
0x0010	LSADC_CTRL4	Interrupt enable register	<a href="#">12-135</a>
0x0014	LSADC_CTRL5	Interrupt status register	<a href="#">12-135</a>
0x0018	LSADC_CTRL6	Interrupt clear register	<a href="#">12-136</a>
0x001C	LSADC_CTRL7	Start configuration register	<a href="#">12-137</a>
0x0020	LSADC_CTRL8	Stop configuration register	<a href="#">12-137</a>
0x0024	LSADC_CTRL9	Conversion result precision register	<a href="#">12-137</a>
0x0028	LSADC_CTRL10	lsadc_zero register	<a href="#">12-138</a>
0x002C	LSADC_CTRL11	LSADC channel 0 data register	<a href="#">12-138</a>
0x0030	LSADC_CTRL12	LSADC channel 1 data register	<a href="#">12-138</a>
0x0034	LSADC_CTRL13	LSADC channel 2 data register	<a href="#">12-138</a>
0x0038	LSADC_CTRL14	LSADC channel 3 data register	<a href="#">12-139</a>
0x0050	LSADC_CTRL20	Average algorithm data register for LSADC channel 0	<a href="#">12-138</a>
0x0054	LSADC_CTRL21	Average algorithm data register for LSADC channel 1	<a href="#">12-139</a>
0x0058	LSADC_CTRL22	Average algorithm data register for LSADC channel 2	<a href="#">12-140</a>
0x005C	LSADC_CTRL23	Average algorithm data register for LSADC channel 3	<a href="#">12-140</a>

## 12.7.5 Register Description

### LSADC\_CTRL0

LSADC\_CTRL0 is an LSDAC configuration register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_800F

Bits	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x00



Bits	Access	Name	Description	Reset
[28:25]	RW	channel_scan_time	Anti-crosstalk period select Single-clock cycle: 333.3 ns 0x0: 15 clock cycles 0x1: 20 clock cycles 0x2: 25 clock cycles 0x3: 30 clock cycles 0x4: 35 clock cycles 0x5: 40 clock cycles 0x6: 45 clock cycles 0x7: 50 clock cycles 0x8: 60 clock cycles 0x9: 70 clock cycles 0xa: 80 clock cycles 0xb: 100 clock cycles 0xc: 150 clock cycles Other values: 15 clock cycles	0x0
[24]	RW	channel_scan_sel	Anti-crosstalk enable bit[1] 0: disabled 1: enabled	0x0
[23:20]	RW	lsadc_data_delta	Error range of the LSADC conversion result. In continuous scanning mode, if the difference between two conversion results falls within the error range, the two results are considered as the same.	0x0
[19:18]	-	reserved	Reserved	0x0
[17]	RW	deglitch_bypass	Deglitch bypass (used in continuous scanning mode) 0: enabled 1: disabled	0x0
[16]	RW	channel_scan_mod_e_sel	Anti-crosstalk enable bit[0] 0: disabled 1: enabled	0x0
[15]	RW	lsadc_reset	Whether the LSADC enters the reset status 0: The LSADC exits the reset status.	0x1



Bits	Access	Name	Description	Reset
			1: The LSADC enters the reset status.	
[14]	-	reserved	Reserved	0x0
[13]	RW	model_sel	Scanning mode of the LSADC 0: single scanning mode 1: continuous scanning mode	0x0
[12]	RW	equ_model_sel	LSADC average algorithm mode select 0: 16-time average algorithm 1: 32-time average algorithm	0x0
[11]	RW	ch_3_vld	LSADC channel 3 validity 0: invalid 1: valid	0x0
[10]	RW	ch_2_vld	LSADC channel 2 validity 0: invalid 1: valid	0x0
[9]	RW	ch_1_vld	LSADC channel 1 validity 0: invalid 1: valid	0x0
[8]	RW	ch_0_vld	LSADC channel 0 validity 0: invalid 1: valid	0x0
[7:4]	RW	reservedcnt_cs_time	System continuous scanning period select in reserved anti-crosstalk mode Single-clock cycle: 333.3ns 0x0: 20 clock cycles 0x1: 25 clock cycles 0x2: 30 clock cycles 0x3: 35 clock cycles 0x4: 40 clock cycles 0x5: 45 clock cycles 0x6: 50 clock cycles 0x7: 60 clock cycles 0x8: 70 clock cycles 0x9: 80 clock cycles 0xa: 100 clock cycles	0x0



Bits	Access	Name	Description	Reset
			0xb: 150 clock cycles 0xc: 200 clock cycles Other values: 20 clock cycles	
[3]	RW	ch_3_eq_enable	Whether the average algorithm for sampling of LSADC channel 3 is valid 0: invalid 1: valid	0x1
[2]	RW	ch_2_eq_enable	Whether the average algorithm for sampling of LSADC channel 2 is valid 0: invalid 1: valid	0x1
[1]	RW	ch_1_eq_enable	Whether the average algorithm for sampling of LSADC channel 1 is valid 0: invalid 1: valid	0x1
[0]	RW	ch_0_eq_enable	Whether the average algorithm for sampling of LSADC channel 0 is valid 0: invalid 1: valid	0x1

## LSADC\_CTRL1

LSADC\_CTRL1 is a deglitch configuration register

Offset Address: 0x0004 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	glitch_sample	LSADC deglitch time window (in ms typically). When the LSADC conversion result is retained in the deglitch time window, the conversion result is considered as a valid value; otherwise, a glitch is considered. The deglitch time window cannot be 0 in continuous scanning mode.	0x0000_0000



## LSADC\_CTRL2

LSADC\_CTRL2 is a scanning interval configuration register.

Offset Address: 0x0008 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	time_scan	Time interval of scanning two channels in continuous scanning mode Tscan = time_scan x 0.333 $\mu$ s (The time interval in continuous scanning mode should be larger than the conversion time of the LSADC, that is, larger than 0x20 cycles.)	0x0000_0000

## LSADC\_CTRL4

LSADC\_CTRL4 is an interrupt enable register.

Offset Address: 0x0010 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:1]	RO	reserved	Reserved	0x0000_0000
[0]	RW	int_enable	Scanned value valid interrupt enable 0: disabled 1: enabled	0x0

## LSADC\_CTRL5

LSADC\_CTRL5 is an interrupt status register.

Offset Address: 0x0014 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved Writing to this field has no effect and reading this field returns 0.	0x0000_0000
[4]	RO	lsadc_auto_busy	LSADC busy indicator in automatic scanning mode 0: idle	0x0



Bits	Access	Name	Description	Reset
			1: busy	
[3]	RO	int_flag_in3	Scanned value validity interrupt flag of channel 3 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	RO	int_flag_in2	Scanned value validity interrupt flag of channel 2 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	int_flag_in1	Scanned value validity interrupt flag of channel 1 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	int_flag_in0	Scanned value validity interrupt flag of channel 0 0: No interrupt is generated. 1: An interrupt is generated.	0x0

## LSADC\_CTRL6

LSADC\_CTRL6 is an interrupt clear register.

Offset Address: 0x0018 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x00000000
[3]	WO	clr_int_flag_in3	Channel 3 interrupt clear 0: not cleared 1: cleared	0x0
[2]	WO	clr_int_flag_in2	Channel 2 interrupt clear 0: not cleared 1: cleared	0x0
[1]	WO	clr_int_flag_in1	Channel 1 interrupt clear 0: not cleared 1: cleared	0x0



Bits	Access	Name	Description	Reset
[0]	WO	clr_int_flag_in0	Channel 0 interrupt clear 0: not cleared 1: cleared	0x0

## LSADC\_CTRL7

LSADC\_CTRL7 is a start configuration register.

Offset Address: 0x001C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	WO	start	LSADC start signal. Writing any value to this register starts the LSADC.	0x0000_0000

## LSADC\_CTRL8

LSADC\_CTRL8 is a stop configuration register.

Offset Address: 0x0020 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	WO	stop	Automatic scanning stop. In automatic scanning mode, writing any value to this register stops automatic scanning of the LSADC. Automatic scanning is restarted only after the start bit is enabled.	0x0000_0000

## LSADC\_CTRL9

LSADC\_CTRL9 is a conversion result precision register.

Offset Address: 0x0024 Total Reset Value: 0x0000\_03FF

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x00000000
[9:0]	RW	lsadc_active_bit	LSADC conversion result precision. 10'b1111111111: 10-bit precision 10'b1111111110: 9-bit precision	0x3FF



Bits	Access	Name	Description	Reset
			... 10'b1000000000: 1-bit precision	

## LSADC\_CTRL10

LSADC\_CTRL10 is an lsadc\_zero register.

Offset Address: 0x0028 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x0000000
[9: 0]	RW	lsadc_zero	LSADC value when no key is pressed	0x000

## LSADC\_CTRL11

LSADC\_CTRL11 is LSADC channel 0 data register.

Offset Address: 0x002C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x0000000
[9:0]	RO	lsadc_data_ina	Scanned value of LSADC channel 0	0x000

## LSADC\_CTRL12

LSADC\_CTRL12 is LSADC channel 1 data register.

Offset Address: 0x0030 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x0000000
[9:0]	RW	lsadc_data_in1	Scanned value of LSADC channel 1	0x000

## LSADC\_CTRL13

LSADC\_CTRL13 is LSADC channel 2 data register.



Offset Address: 0x0034 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x000000
[9:0]	RO	lsadc_data_in0	Scanned value of LSADC channel 2	0x000

## LSADC\_CTRL14

LSADC\_CTRL14 is LSADC channel 3 data register.

Offset Address: 0x0038 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x000000
[9:0]	RO	lsadc_data_in0	Scanned value of LSADC channel 3	0x000

## LSADC\_CTRL20

LSADC\_CTRL20 is the LSADC channel 0 average algorithm data register.

Offset Address: 0x0050 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15]	RO	lsadc_equ_value_valid_0	Whether the average algorithm for sampling of LSADC channel 0 is valid 0: invalid 1: valid	0x0
[11:10]	-	reserved	Reserved	0x0
[9:0]	RO	lsadc_equ_value_0	Result of the average algorithm for sampling of LSADC channel 0	0x000

## LSADC\_CTRL21

LSADC\_CTRL21 is the average algorithm data register for LSADC channel 1.

Offset Address: 0x0054 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15]	RO	lsadc_equ_value_valid_1	Whether the average algorithm for sampling of LSADC channel 1 is valid 0: invalid 1: valid	0x0
[11:10]	-	reserved	Reserved	0x0
[9:0]	RO	lsadc_equ_value_1	Result of the average algorithm for sampling of LSADC channel 1	0x000

## LSADC\_CTRL22

LSADC\_CTRL22 is the average algorithm data register for LSADC channel 2.

Offset Address: 0x0058 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15]	RO	lsadc_equ_value_valid_2	Whether the average algorithm for sampling of LSADC channel 2 is valid 0: invalid 1: valid	0x0
[11:10]	-	reserved	Reserved	0x0
[9:0]	RO	lsadc_equ_value_2	Result of the average algorithm for sampling of LSADC channel 2	0x000

## LSADC\_CTRL23

LSADC\_CTRL23 is the average algorithm data register for LSADC channel 3.

Offset Address: 0x005C Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15]	RO	lsadc_equ_value_valid_3	Whether the average algorithm for sampling of LSADC channel 3 is valid 0: invalid	0x0



Bits	Access	Name	Description	Reset
			1: valid	
[11:10]	-	reserved	Reserved	0x0
[9:0]	RO	lsadc_equ_value_3	Result of the average algorithm for sampling of LSADC channel 3	0x000

## 12.8 PWM0

### 12.8.1 Overview

The chip has one programmable PWM0 controller, which provides four outputs corresponding to the SVB\_PWM, PWM0\_OUT1, PWM0\_OUT2, and PWM0\_OUT3 pins.

#### WARNING

- The SVB\_PWM pin is dedicated for system voltage scaling. Unauthorized configuration of related registers may cause SVB exceptions.
- PWM0 does not support the configuration of the working clock, reset, and number of output pulses. It supports only the configuration of the output period, duty cycle, and polarity. Therefore, PWM1 is recommended.

For the PWM0 controller output:

- The working clock is 50 MHz.
- The output period is configurable. A maximum of 25 MHz (50 MHz/2) output and a minimum of 50 kHz (50 MHz/1000) output are supported.
- The number of output high-level clock cycles is configurable.
- The duty cycle counter is 10 bits. The PWM0 output with the specified period and duty cycle can be implemented by configuring PWM0 registers.

### 12.8.2 Configuration Process

The PWM0\_OUT(n) output configuration process is as follows:

- Step 1** Calculate the PWM period and the number of high level times in the PWM period based on the required PWM output frequency (freq) and duty cycle (duty).

The formula for calculating the number of periods is as follows:

$$\text{pwm0\_out}(n)\text{period}=(50000000/\text{Freq})-1$$



The formula for calculating the high level times in the PWM period is as follows:

$$\text{pwm0\_out}(n)\text{duty} = (50000000/\text{Freq}) * \text{duty} - 1$$

- Step 2** Set the number of PWM periods ( $\text{pwm0\_out}(n)\text{period}$ ) and the number of high-level times in the PWM period ( $\text{pwm0\_out}(n)\text{duty}$ ).
- Step 3** Set  $\text{PWM0\_OUT}(n)\text{CTRL}[2]$  to 1 to load the configured parameter and validate it.
- Step 4** Set  $\text{PWM0\_OUT}(n)\text{CTRL}[0]$  to 1 to enable the PWM output.

#### NOTE

In the preceding steps, **50000000** is the working clock, and **n** is **1**, **2**, or **3**, corresponding to three  $\text{PWM0\_OUT}$  channels. You can make the settings in steps 2 to 4 together.

----End

For example,  $\text{PWM0\_OUT1}$  needs to output a waveform whose frequency is 50 kHz and high level occupies 72.5% (duty cycle).

The number of PWM periods is  $50\text{ MHz}/50\text{ kHz} = 1000$  (rounded off), which is 0x3E8 in hexadecimal format. The value is 0x3E7 after being subtracted by 1 according to step 1. The number of high-level times is  $1000$  (number of PWM periods)  $\times$  72.5% (duty cycle) = 725 (rounded off), which is 0x2D5 in hexadecimal format. The value is 0x2D4 after being subtracted by 1 according to step 1.

After the calculation is complete, configure the following:

- $\text{PWM0\_OUT1\_CTRL}[25:16] = 0x2D4$
- $\text{PWM0\_OUT1\_CTRL}[13:4] = 0x3E7$
- $\text{PWM0\_OUT1\_CTRL}[2] = 1$
- $\text{PWM0\_OUT1\_CTRL}[0] = 1$

That is, set  $\text{PWM0\_OUT1}$  to 0x02D43E75 to output the required waveform.

### 12.8.3 Summary of PWM0 Registers

Table 12-19 describes the PWM0 registers.

**Table 12-19** Summary of PWM0 registers (base address: 0x1102\_9000)

Offset Address	Register	Description	Page
0x0000	SVB_PWM_CTRL	SVB_PWM control register	<a href="#">12-143</a>
0x0004	PWM0_OUT1_CTRL	PWM0_OUT1 control register	<a href="#">12-143</a>
0x0008	PWM0_OUT2_CTRL	PWM0_OUT 2 control register	<a href="#">12-144</a>
0x000C	PWM0_OUT3_CTRL	PWM0_OUT 3 control register	<a href="#">12-144</a>



## 12.8.4 Description of PWM0 Registers

### SVB\_PWM\_CTRL

SVB\_PWM\_CTRL is the SVB\_PWM control register.

Offset Address: 0x0000 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:0]	-	reserved	Reserved This register is a special-purpose register for system voltage scaling. Do not configure it; otherwise, the system voltage becomes abnormal.	0x00

### PWM0\_OUT1\_CTRL

PWM0\_OUT1\_CTRL is the PWM0\_OUT1 control register.

Offset Address: 0x0004 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RW	pwm0_out1_duty	Number of high-level times in a PWM period of the PWM0_OUT1 output. The actual number is the configured value plus 1.	0x000
[15:14]	-	reserved	Reserved	0x0
[13:4]	RW	pwm0_out1_perio d	PWM period of the PWM0_OUT1 output. The actual number is the configured value plus 1.	0x000
[3]	-	reserved	Reserved	0x0
[2]	W1_PUL SE	pwm0_out1_load	Parameter load control of the PWM0_OUT1 output. Writing 1 to this bit enables the parameter to take effect.	0x0
[1]	RW	pwm0_out1_inv	Phase control of the PWM0_OUT1 output 0: positive phase 1: inversed phase	0x0



Bits	Access	Name	Description	Reset
[0]	RW	pwm0_out1_enable	PWM0_OUT1 enable 0: disabled 1: enabled	0x0

## PWM0\_OUT2\_CTRL

PWM0\_OUT2\_CTRL is the PWM0\_OUT2 control register.

Offset Address: 0x0008 Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RW	pwm0_out2_duty	Number of high-level times in a PWM period of the PWM0_OUT2 output. The actual number is the configured value plus 1.	0x000
[15:14]	-	reserved	Reserved	0x0
[13:4]	RW	pwm0_out2_period	PWM period of the PWM0_OUT2 output. The actual number is the configured value plus 1.	0x000
[3]	-	reserved	Reserved	0x0
[2]	W1_PULSE	pwm0_out2_load	Parameter load control of the PWM0_OUT2 output. Writing 1 to this bit enables the parameter to take effect.	0x0
[1]	RW	pwm0_out2_inv	Phase control of the PWM0_OUT2 output 0: positive phase 1: inversed phase	0x0
[0]	RW	pwm0_out2_enable	PWM0_OUT2 enable 0: disabled 1: enabled	0x0

## PWM0\_OUT3\_CTRL

PWM0\_OUT3\_CTRL is the PWM0\_OUT3 control register.

Offset Address: 0x000C Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25:16]	RW	pwm0_out3_duty	Number of high-level times in a PWM period of the PWM0_OUT3 output. The actual number is the configured value plus 1.	0x000
[15:14]	-	reserved	Reserved	0x0
[13:4]	RW	pwm0_out3_period	PWM period of the PWM0_OUT3 output. The actual number is the configured value plus 1.	0x000
[3]	-	reserved	Reserved	0x0
[2]	W1_PULSE	pwm0_out3_load	Parameter load control of the PWM0_OUT3 output. Writing 1 to this bit enables the parameter to take effect.	0x0
[1]	RW	pwm0_out3_inv	Phase control of the PWM0_OUT3 output 0: positive phase 1: inversed phase	0x0
[0]	RW	pwm0_out3_enable	PWM0_OUT3 enable 0: disabled 1: enabled	0x0

## 12.9 PWM1

### 12.9.1 Overview

The pulse width modulation (PWM1) module outputs periodic pulse signals.

The chip provides one group of PWM1 controllers:

Provides a maximum of eight PWM signal outputs, corresponding to PWM1\_OUT0–7.

### 12.9.2 Features

The PWM1 controller supports the following features:

- Working clock

The PWM1 can be switched between 198 MHz, 24 MHz, and 1 MHz (for details, see PERI\_CRG4454 in chapter "System").

- Each PWM output can be separately enabled, for example, PWM1\_OUT3.
- Supports periodic output of square waves. The period is configurable.
- Supports a fixed number of square wave outputs. The number of square waves is configurable.
- Supports left alignment, right alignment, and center alignment of pulse signals.
- The polarity of pulse signals is controllable.
- Supports dynamic update of the period, duty cycle, alignment mode, and polarity.
- Multiple PWM signals can be output synchronously in single or periodic mode.
- The duty cycle can be set to 0% (low level) to 100% (high level). For details about the configuration process, see section [12.9.4 "Configuration Process."](#)

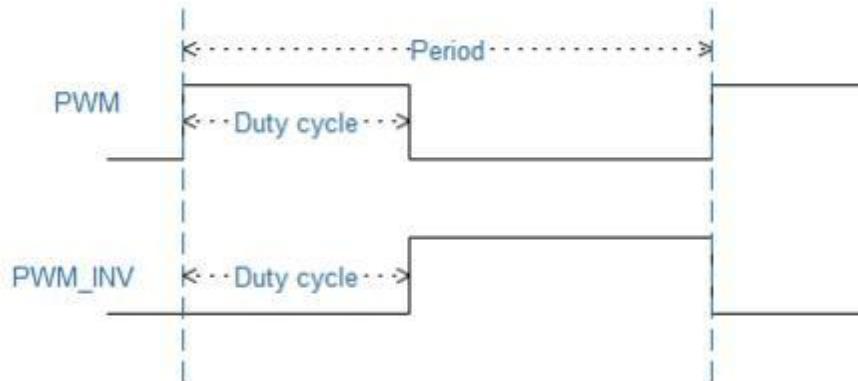
## 12.9.3 Operating Mode

### 12.9.3.1 Alignment Mode

The pulse signals output by the PWM1 support three modes: left-aligned, right-aligned, and center-aligned. You can set the pulse alignment mode of each PWM output by configuring [PWM\(n\)\\_ctrl.pwm\(n\)\\_align\\_mode\\_cfg](#).

#### Left-Aligned Mode

**Figure 12-38** Left-aligned output waveform



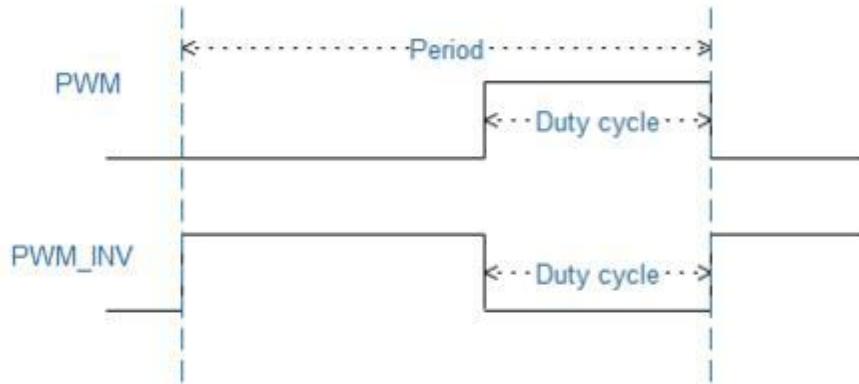
Set [PWM\(n\)\\_ctrl.pwm\(n\)\\_align\\_mode\\_cfg](#) to 1 to enable the left-aligned mode.

In this mode, the PWM outputs the valid signal level based on the duty cycle configuration, and then outputs the invalid signal level until the configured cycle output is complete.

The valid signal level is high by default and can be controlled by configuring [PWM\(n\)\\_ctrl.pwm\(n\)\\_inv\\_cfg](#). When [PWM\(n\)\\_ctrl.pwm\(n\)\\_inv\\_cfg](#) is set to 1, the valid signal level is low, and the corresponding waveform is PWM\_INV.

## Right-Aligned Mode

Figure 12-39 Right-aligned output waveform



Set `PWM(n)_ctrl.pwm(n)_align_mode_cfg` to **0** to enable the right-aligned mode. In this mode, the PWM outputs the invalid signal level first, and then outputs the valid signal level based on the duty cycle configuration.

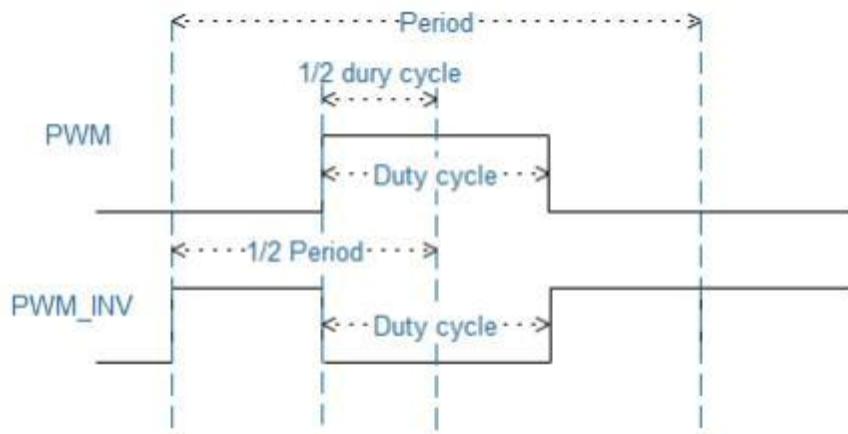
The valid signal level is high by default and can be controlled by configuring `PWM(n)_ctrl.pwm(n)_inv_cfg`. When `PWM(n)_ctrl.pwm(n)_inv_cfg` is set to **1**, the valid signal level is low, and the corresponding waveform is `PWM_INV`.

## Center-Aligned Mode

### NOTICE

The duty cycle and the number of cycles can only be even numbers in center-aligned mode.

Figure 12-40 Center-aligned output waveform



Set `PWM(n)_ctrl.pwm(n)_align_mode_cfg` to **2** to enable the center alignment mode.

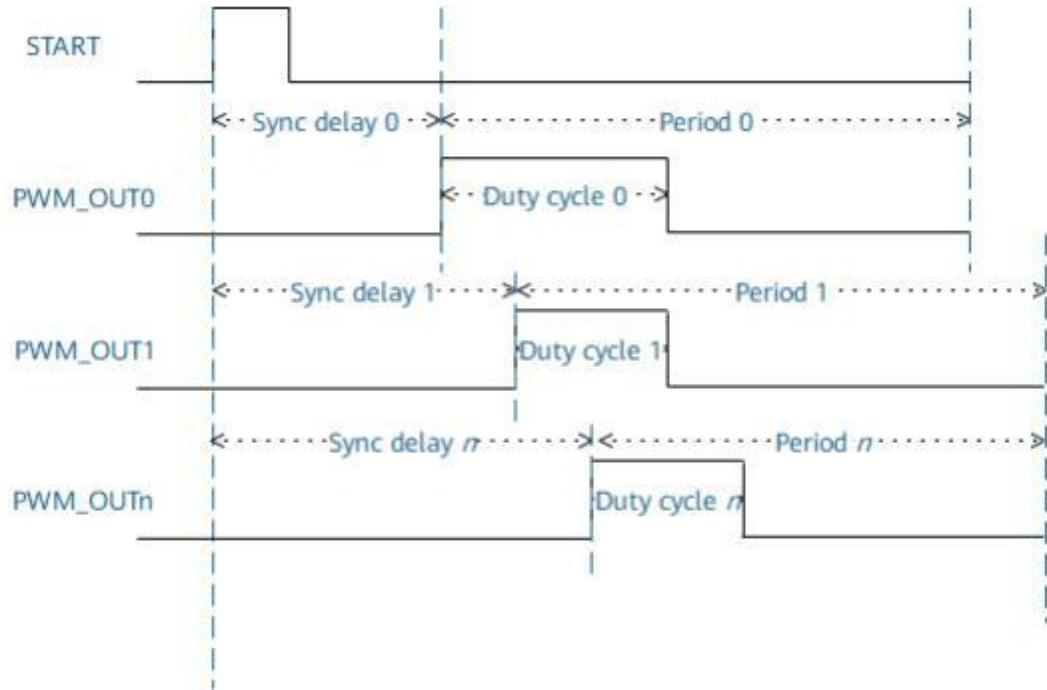
The PWM outputs the invalid signal level, outputs the valid signal level based on the duty cycle configuration, and then outputs the invalid signal level. During the entire period, the valid signal level is in the center.

The valid signal level is high by default and can be controlled by configuring `PWM(n)_ctrl.pwm(n)_inv_cfg`. When `PWM(n)_ctrl.pwm(n)_inv_cfg` is set to **1**, the valid signal level is low, and the corresponding waveform is `PWM_INV`.

### 12.9.3.2 Sync Mode

The multiple PWM signals provided by each PWM controller are output synchronously. The sync delay of each channel can be configured independently. See [Figure 12-41](#).

**Figure 12-41** Output waveforms in sync mode



From the START signal, each PWM starts after a delay based on the configuration of `PWM(n)_sync_delay_cfg`. The period, duty cycle, and alignment mode of each PWM can be configured independently.

In PWM sync mode, the trigger mode and cyclic output mode are supported.

- Trigger mode (output of several pulse signals)

After the number of pulse output signals is configured for each PWM output signal, the output stops until the next START.



Note: In trigger mode, you need to set the `PWM(n)_sync_cfg.pwm(n)_sync_mux` synchronization source to a non-operating PWM output.

- Cyclic mode (periodic output of pulse signals)  
A sync source is configured for each PWM by using `PWM(n)_sync_cfg.pwm(n)_sync_mux`. After the PWM signal corresponding to the sync source is output, each PWM is reloaded and enters the START state again to implement periodic output.  
Note: When the sync delay (`PWM(n)_sync_delay_cfg`) is set in cyclic mode, ensure that the delay of the `PWM(n)_sync_cfg.pwm(n)_sync_mux` synchronization source is the minimum.

## 12.9.4 Configuration Process

### 12.9.4.1 General Mode

The configuration process is as follows:

- Step 1** Enable clock gating by configuring PERI\_CRG4454.
- Step 2** Calculate the required number of periods and high level cycles and write them to the registers `PWM(n)_duty0_cfg` and `PWM(n)_period`.
- Step 3** To output pulse signals periodically, set `PWM(n)_ctrl[2]` to **1**. To output a fixed number of pulse signals, set `PWM(n)_ctrl[2]` to **1** and write the number of pulses to `PWM(n)_num_cfg`.
- Step 4** Configure the register `PWM(n)_ctrl[5:4]` according to the alignment mode of the output pulse signal.
- Step 5** Configure the register `PWM(n)_ctrl[1]` based on the polarity of the output pulse signal.
- Step 6** Write **1** to the pulse enable register `PWM(n)_ctrl[0]` to enable the PWM output.

----End

Assume that the output waveform is 4 kHz and has 80% (duty cycle) high levels and 10 right-aligned pulses.

The number of periods is **49500**, which is rounded off by 198 MHz x 1/4 kHz, or **0x000C15C** in hexadecimal format. The number of high levels is **39600**, which is rounded off by 49500 (number of periods) x 80% (duty cycle), or **0x00009AB0** in hexadecimal format.

To output required waveforms, perform the following steps:

- Step 1** Write **0x1** to the corresponding CRG register (for example, PERI\_CRG4454[4]) to enable the PWM clock.
- Step 2** Set `PWM(n)_ctrl` to **0x100**.
- Step 3** Write **0x000C15B** to `PWM(n)_period_cfg`.



- Step 4** Write **0x00009AAF** to **PWM(n)\_duty0\_cfg**.
  - Step 5** Write **0x9** to **PWM(n)\_num\_cfg**.
  - Step 6** Write **0x101** to **PWM(n)\_ctrl** to update only bit[0].
  - Step 7** Read **PWM(n)\_ctrl\_st** bit[0] and wait until bit[10] is **1** (indicating that the PWM is outputting square waves).
  - Step 8** Read **PWM(n)\_period** and compare the read value with **0x000C15B**.
  - Step 9** Read **PWM(n)\_duty0** and compare the read value with **0x00009AAF**.
  - Step 10** Read **PWM(n)\_num** bit[15:0] and compare the read value with **0x09**. If **PWM(n)\_ctrl\_st** bit[0] is **1**, the PWM module is outputting square waves. If **PWM(n)\_ctrl\_st** bit[10] is **0**, a configured number of square waves have been output.
- End

#### 12.9.4.2 Methods for Configuring 0% and 100% Duty Cycles

##### 100% Duty Cycle

Assume that one high level needs to be output.

The number of periods is **49500**, which is rounded off by  $198 \text{ MHz} \times 1/4 \text{ kHz}$ , or **0x0000C15C** in hexadecimal format. The number of high levels is **49500**, which is rounded off by  $49500 \text{ (number of periods)} \times 100\% \text{ (duty cycle)}$ , or **0x0000C15C** in hexadecimal format.

To output required waveforms, perform the following operations:

- Step 1** Write **0x1** to the corresponding CRG register (for example, PERI\_CRG4454[4]) to enable the PWM clock.
  - Step 2** Set **PWM(n)\_ctrl** to **0x104**.
  - Step 3** Write **0x0000C15B** to **PWM(n)\_period\_cfg**.
  - Step 4** Write **0x0000C15B** to **PWM(n)\_duty0\_cfg**.
  - Step 5** Write **0x105** to **PWM(n)\_ctrl** to update only bit[0].
- End

##### 0% Duty Cycle

Assume that one low level needs to be output.

The number of periods is **49500**, which is rounded off by  $198 \text{ MHz} \times 1/4 \text{ kHz}$ , or **0x0000C15C** in hexadecimal format. The number of high levels is **49500**, which is rounded off by  $49500 \text{ (number of periods)} \times 100\% \text{ (duty cycle)}$ , or **0x0000C15C** in hexadecimal format.



To output required waveforms, perform the following operations:

- Step 1** Write **0x1** to the corresponding CRG register (for example, PERI\_CRG4454[4]) to enable the PWM clock.
- Step 2** Set **0x106** to [PWM\(n\)\\_ctrl](#). (Set the PWM positive phase output to 100% duty cycle, that is, the PWM negative phase output to 0% duty cycle.) In this example, [pwm\(n\)\\_inv\\_cfg](#) is set to **1**, and the PWM output phase is inverted.
- Step 3** Write **0x0000C15B** to [PWM\(n\)\\_period\\_cfg](#).
- Step 4** Write **0x0000C15B** to [PWM\(n\)\\_duty0\\_cfg](#).
- Step 5** Write **0x107** to [PWM\(n\)\\_ctrl](#) to update only bit[0].

----End

#### 12.9.4.3 Sync Mode

The configuration process is as follows:

- Step 1** Configure the PWM sync mode by setting [PWM\(n\)\\_sync\\_cfg](#)bit[5] to **0x1**.

- Step 2** Configure the register based on the trigger mode or cyclic mode.

- Trigger mode: Set [PWM\(n\)\\_sync\\_cfg](#)bit[4] to **0**.  
Note: In trigger mode, you need to set the [PWM\(n\)\\_sync\\_cfg](#).pwm(n)\_sync\_mux synchronization source to a non-operating PWM output.
- Cyclic mode: Set [PWM\(n\)\\_sync\\_cfg](#)bit[5] to **1**. Configure the PWM reload signal source as required by setting [PWM\(n\)\\_sync\\_cfg](#)bit[3:0].  
Note: When the sync delay ([PWM\(n\)\\_sync\\_delay\\_cfg](#)) is set in cyclic mode, ensure that the delay of the [PWM\(n\)\\_sync\\_cfg](#).pwm(n)\_sync\_mux synchronization source is the minimum.

- Step 3** Configure the PWM signal mode (the same as steps 1–8 in general mode).

- Step 4** Enable the PWM by configuring the corresponding bit of [PWM\\_sync\\_start](#).

For example, if PWM0, PWM1, and PWM5 need to be started synchronously, write **0x23** to [PWM\\_sync\\_start](#).

----End

#### 12.9.5 Summary of Variables in the Register Offset Addresses

PWM1: The base address is **0x1108\_0000**.

[Table 12-20](#)describes the value range and meaning of the variable in the register offset address.



**Table 12-20** Variable in the register offset address

Variable	Value Range	Description
n	0-7	8 channels of the PWM module

## 12.9.6 Register Summary

Table 12-21 describes the PWM1 registers.

**Table 12-21** Summary of PWM registers

Offset Address	Name	Description	Page
0x0000 + 0x100×n	PWM(n)_period_cfg	PWM(n) period configuration register	<a href="#">12-153</a>
0x0004 + 0x100×n	PWM(n)_duty0_cfg	PWM(n) duty cycle configuration register	<a href="#">12-153</a>
0x0010 + 0x100×n	PWM(n)_num_cfg	PWM(n) output square wave quantity configuration register	<a href="#">12-153</a>
0x0014 + 0x100×n	PWM(n)_ctrl	PWM(n) working mode configuration register	<a href="#">12-154</a>
0x0030 + 0x100×n	PWM(n)_sync_cfg	PWM(n) sync mode configuration register	<a href="#">12-154</a>
0x0034 + 0x100×n	PWM(n)_sync_delay_cfg	PWM(n) sync delay configuration register	<a href="#">12-155</a>
0x0040 + 0x100×n	PWM(n)_period	PWM(n) real-time period readback register	<a href="#">12-155</a>
0x0044 + 0x100×n	PWM(n)_duty0	PWM(n) real-time duty cycle 0 readback register	<a href="#">12-156</a>
0x0050 + 0x100×n	PWM(n)_num	PWM(n) real-time square wave quantity readback register.	<a href="#">12-156</a>
0x0054 + 0x100×n	PWM(n)_ctrl_st	PWM(n) real-time working status readback register	<a href="#">12-156</a>
0x0074 + 0x100×n	PWM(n)_sync_delay	PWM(n) real-time sync delay configuration readback register	<a href="#">12-157</a>
0xFF0	PWM_sync_start	PWM sync start enable register	<a href="#">12-157</a>



## 12.9.7 Register Description

### PWM(n)\_period\_cfg

PWM(n)\_period\_cfg is the PWM(n) period configuration register.

Offset Address: 0x0000 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19:0]	RW	pwm(n)_period_cfg	PWM(n) period configuration. Unit: clock cycle. The actual value is the configured value minus 1.	0x00000

### PWM(n)\_duty0\_cfg

PWM(n)\_duty0\_cfg is the PWM(n) duty cycle configuration register.

Offset Address: 0x0004 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19:0]	RW	pwm(n)_duty0_cfg	Duty cycle configuration of the PWM(n) valid signal. Unit: clock cycle. The actual value is the configured value minus 1.	0x00000

### PWM(n)\_num\_cfg

PWM(n)\_num\_cfg is the PWM(n) output square wave quantity configuration register.

Offset Address: 0x0010 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	pwm(n)_num_cfg	Number of square waves output by PWM(n). The actual value is the configured value minus 1.  This parameter is valid only when <b>pwm(n)_keep</b> is set to <b>0</b> .	0x0000



## PWM(n)\_ctrl

PWM(n)\_ctrl is the PWM(n) working mode configuration register.

Offset Address: 0x0014 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:4]	RW	pwm(n)_align_mode_cfg	PWM(n) alignment mode control 00: right-aligned mode 01: left-aligned mode 10: center-aligned mode Other values: reserved	0x0
[3]	-	reserved	Reserved	0x0
[2]	RW	pwm(n)_keep_cfg	PWM(n) output mode control 0: <b>pwm(n)_num</b> square waves are output. 1: Square waves are always output.	0x0
[1]	RW	pwm(n)_inv_cfg	PWM(n) output phase control 0: forward 1: inverted	0x0
[0]	RW	pwm(n)_enable	PWM(n) enable 0: disabled 1: enabled	0x0

## PWM(n)\_sync\_cfg

PWM(n)\_sync\_cfg is the PWM(n) sync mode configuration register.

Offset Address: 0x0030 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5]	RW	pwm(n)_sync_enable	PWM sync working mode enable 0: normal mode 1: sync working mode. PWM_SYNC_START starts the PWM	0x0



Bits	Access	Name	Description	Reset
			function.	
[4]	RW	pwm(n)_sync_mod_e	PWM sync mode select 0: trigger mode 1: cyclic mode	0x0
[3:0]	RW	pwm(n)_sync_mux	Reload signal source select in cyclic mode 0x0: PWM0 completion signal 0x1: PWM1 completion signal 0x2: PWM2 completion signal 0x3: PWM3 completion signal 0x4: PWM4 completion signal 0x5: PWM5 completion signal 0x6: PWM6 completion signal 0x7: PWM7 completion signal Others: reserved	0x0

## PWM(n)\_sync\_delay\_cfg

PWM(n)\_sync\_delay\_cfg is the PWM(n) sync delay configuration register.

Offset Address: 0x0034 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19:0]	RW	pwm(n)_sync_delay_cfg	Start delay configuration register for PWM(n) sync mode. Unit: period. The actual value is the configured value minus 1.	0x00000

## PWM(n)\_period

PWM(n)\_period is the PWM(n) real-time period readback register.

Offset Address: 0x0040 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000



Bits	Access	Name	Description	Reset
[19:0]	RO	pwm(n)_period	Effective value of the PWM(n) period	0x00000

## PWM(n)\_duty0

PWM(n)\_duty0 is the PWM(n) real-time duty cycle 0 readback register.

Offset Address: 0x0044 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19:0]	RO	pwm(n)_duty0	Effective value of PWM(n) duty cycle 0	0x00000

## PWM(n)\_num

PWM(n)\_num is the PWM(n) real-time square wave quantity readback register.

Offset Address: 0x0050 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:16]	RO	pwm(n)_num_cnt	Number of square waves to be output by PWM(n). This parameter is valid only when <b>pwm(n)_busy</b> is 1 and <b>pwm(n)_keep</b> is 0.	0x0000
[15:0]	RO	pwm(n)_num	Effective value of the number of square waves output by PWM(n)	0x0000

## PWM(n)\_ctrl\_st

PWM(n)\_ctrl\_st is the PWM(n) real-time working status readback register.

Offset Address: 0x0054 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x00000000
[5:4]	RO	pwm(n)_align_mode	Effective value of the alignment mode of PWM(n)	0x0



Bits	Access	Name	Description	Reset
			00: right-aligned mode 01: left-aligned mode 10: center-aligned mode Other values: reserved	
[3]	-	reserved	Reserved	0x0
[2]	RO	pwm(n)_keep	Effective value of the output mode of PWM(n) 0: <a href="#">PWM(n)_num</a> square waves are output. 1: Square waves are always output.	0x0
[1]	RO	pwm(n)_inv	Effective value of the output phase of PWM(n) 0: forward 1: inverted	0x0
[0]	RO	pwm(n)_busy	Working status of PWM(n) 0: idle 1: working	0x0

### PWM(n)\_sync\_delay

PWM(n)\_sync\_delay is the PWM(n) real-time sync delay configuration readback register.

Offset Address: 0x0074 + 0x100×n Total Reset Value: 0x0000\_0000

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19:0]	RO	pwm(n)_sync_delay	Effective value of the start delay configuration in PWM(n) sync mode	0x00000

### PWM\_sync\_start

PWM\_sync\_start is the PWM sync start enable register.

Offset Address: 0xFF0 Total Reset Value: 0x0000\_0000



Bits	Access	Name	Description	Reset
[31:8]	-	resverd	Reserved	0x00000000
[7]	WO	pwm7_sync_start	PWM7 start configuration register. Writing 1 starts PWM7.	0x0
[6]	WO	pwm6_sync_start	PWM6 start configuration register. Writing 1 starts PWM6.	0x0
[5]	WO	pwm5_sync_start	PWM5 start configuration register. Writing 1 starts PWM5.	0x0
[4]	WO	pwm4_sync_start	PWM4 start configuration register. Writing 1 starts PWM4.	0x0
[3]	WO	pwm3_sync_start	PWM3 start configuration register. Writing 1 starts PWM3.	0x0
[2]	WO	pwm2_sync_start	PWM2 start configuration register. Writing 1 starts PWM2.	0x0
[1]	WO	pwm1_sync_start	PWM1 start configuration register. Writing 1 starts PWM1.	0x0
[0]	WO	pwm0_sync_start	PWM0 start configuration register. Writing 1 starts PWM0.	0x0



## Contents

<b>13 Security Subsystem .....</b>	<b>13-1</b>
13.1 Overview.....	13-1



# 13 Security Subsystem

## 13.1 Overview

The security subsystem supports the following security features:

- True random numbers generated by hardware, in compliance with the SP800-22 standard
- One-Time Programmable (OTP), with the available user space up to 28 Kbits
- Hardware-based symmetric encryption/decryption algorithms and hash algorithms:
  - Symmetric encryption/decryption algorithms
    - AES128 and AES256, which are implemented by complying with the FIPS 197 standard and working in the modes complying with the NIST special800-38a standard
    - SM4
  - Hash algorithms
    - SHA256, SHA384, SHA512, HMAC-SHA256, HMAC-SHA384, and HMAC-SHA512
    - SM3
- Hardware-based asymmetric encryption/decryption algorithms
  - RSA 3072/4096, in compliance with the PKCS#1 V1.5/2.1 standard
  - SM2
  - ECC P-256, P-384, and P-512
- JTAG protection
- KeyLadder, used for key encryption/decryption management to improve security
- Arm TrustZone
- Secure memory isolation



- Secure boot and signature verification, encryption, and decryption of the bootloader



---

## Contents

---

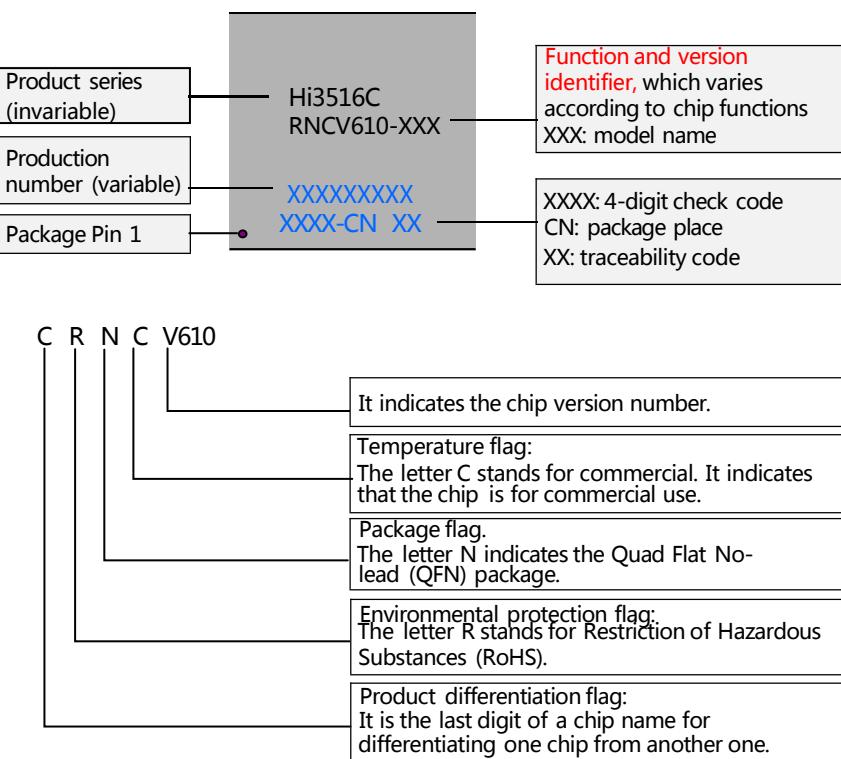
<b>A Ordering Information.....</b>	<b>1</b>
------------------------------------	----------



# A Ordering Information

There are seven Hi3516CV610 models based on functions: 00B, 10B, 20B, 00S, 20S, 00G, and 20G. Hi3516CV610-10B/20B/20S/20G uses the QFN package, while Hi3516CV610-00B/00S/00G uses the TFBGA package. [Figure A-1](#) and [Figure A-2](#) show the chip marks.

**Figure A-1** Hi3516CV610-10B/20B/20S/20G marks

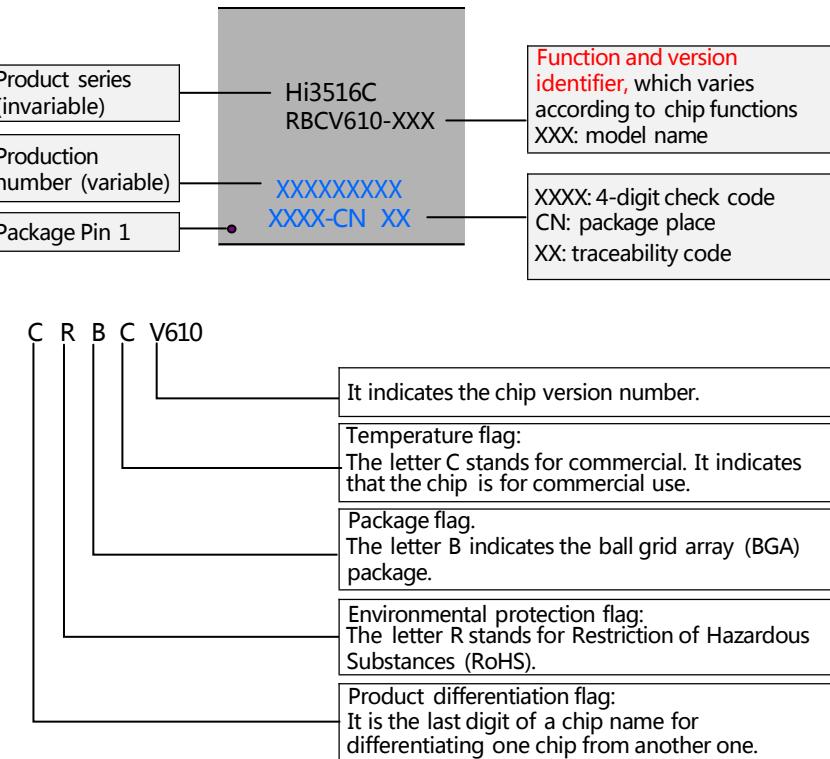




**Table A-1** Hi3516CV610-10B/20B/20S/20G packages

Part Number	Package Type	Package Dimension	Pitch
Hi3516CRNCV610	QFN	9 mm x 9 mm(0.35 in. x 0.35 in.)	0.35 mm (0.03 in.)

**Figure A-2** Hi3516CV610-00B/00S/00G marks



**Table A-2** Hi3516CV610-00B/00S/00G packages

Part Number	Package Type	Package Dimension	Pitch
Hi3516CRBCV610	TFBGA	12 mm x 13.3 mm (0.47 in. x 0.52 in.)	0.65 mm (0.03 in.)



# B

## Acronyms and Abbreviations

3DNR	three-dimensional noise reduction	ABR	average bit rate
AE	automatic exposure	AEC	acoustic echo control
AES	advanced encryption standard	AF	automatic focus
ALC	automatic level control	ANR	adaptive noise reduction
API	application programming interface	AVBR	adaptive variable bit rate
AWB	automatic white balance	CBR	constant bit rate
CMOS	complementary metal-oxide-semiconductor	codec	coder
CV	computer vision	DDR	double data rate
DC	digital camera	DMA	direct memory access
DDRC	double data rate controller	EP	endpoint
eMMC	embedded multimedia card	FPN	fixed pattern noise
FCCSP	flip-chip chip scale package	GE	gigabit Ethernet
FPU	floating-point unit	HiSPI	high-speed serial pixel interface
GPIO	general-purpose input/output	IS	inter-IC sound
IC	inter-integrated circuit	IVE	intelligent video engine
ISP	image signal processor	LSADC	low-speed analog-to-digital converter
LCD	liquid crystal display	LVDS	low-voltage differential signaling
MIC	microphone	MIPI	mobile industry processor interface
MMU	memory management unit	NR	noise reduction
OTP	one-time programming	OSD	on-screen display



PWM	pulse-width modulation	POR	power-on reset
PMC	power management controller	RAM	random access memory
RC	root complex	RGB	red-green-blue
RoHS	restriction of hazardous substances	ROI	region of interest
RMII	reduced media-independent interface	RSA	Rivest-Shamir-Adleman
RTC	Realtime clock	RNG	random number generator
SD	secure digital	SDIO	secure digital input/output
SDRAM	synchronous dynamic random access memory	SDK	software development kit
SDXC	secure digital extended capacity	SoC	system-on-chip
SPI	serial peripheral interface	TDM	time division multiplexing
TOPS	Tera Operations Per Second	TSO	TCP segmentation offload
TX	transmit	UART	universal asynchronous receiver transmitter
USB	Universal Serial Bus	VBR	variable bit rate
VI	video input	VQE	voice quality enhancement
WDR	wide dynamic range		