Hi3515 H.264 Encoding and Decoding Processor

Data Sheet

Issue 02

Date 2010-04-20

HiSilicon Technologies CO., LIMITED.

Address: Manufacture Center of Huawei Electric, Huawei Base,
Bantian, Longgang District, Shenzhen, 518129, People’s Republic of China

Website: http://www.hisilicon.com

Tel: +86-755-28788858

Fax: +86-755-28357515

Email: support@hisilicon.com

About This Document

Purpose

This document describes the features, logic structures, module functions, operating modes, register definitions of the Hi3515. It also describes the interface timings and parameters in figures and tables. In addition, this document provides the pin definitions, performance parameters, and package dimensions.

Related Version

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

Product Name	Version
Hi3515 H.264 Encoding and Decoding Processor	V100

Intended Audience

This document is intended for:

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

Conventions

Symbol Conventions

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

Symbol	Description
 DANGER	Indicates a hazard with a high level of risk which, if not avoided, will result in death or serious injury.

Symbol	Description
WARNING	Indicates a hazard with a medium or low level of risk that, if not avoided, could result in minor or moderate injury.
CAUTION	Indicates a potentially hazardous situation, which if not avoided, could result in equipment damage, data loss, performance degradation, or unexpected results.
TIP	Indicates a tip that may help you solve a problem or save time.
NOTE	Provides additional information to emphasize or supplement important points of the main text.

General Conventions

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

Convention	Description
Times New Roman	Normal paragraphs are in Times New Roman.
Boldface	Names of files, directories, folders, and users are in **boldface**. For example, log in as user **root**.
Italic	Book titles are in *italics*.
`Courier New`	Examples of information displayed on the screen are in `Courier New`.

Table Conventions

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

Convention	Description
–	The cell is blank.
*	The contents in such cell are configurable.

Register Attributes

The register attributions 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.

Symbol	Description	Symbol	Description
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.

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)	1K	1024
	1M	1,048,576
	1G	1,073,741,824
Frequency, data rate	1k	1000
	1M	1,000,000
	1G	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.

Update History

Updates between document issues are cumulative. Therefore, the latest document issue contains all updates made in previous issues.

Updates in Issue 02 (2010-04-20)

Chapter 1 Product Description

In section 1.1.13, the typical power consumption is changed from 1,000 mW to 1,500 mW; the surface temperature of the chip package ranging from –20°C to +110°C is added, the junction temperature ranging from –40°C to +125°C is added.

Chapter 2 Hardware

In Table 2-34, the surface temperature of the chip package ranging from –20°C to +110°C is added and the junction temperature ranging from –40°C to +125°C is added.

Updates in Issue 01 (2010-03-26)

Chapter 2 Hardware

In Table 2-34, the minimum values, typical values, and maximum values of DVDD10, DVDD10_PLL, DVDD10_DAC, and SPHY_VP are changed from 0.95, 1.0, 1.1 to 1.0, 1.05, 1.15 respectively; the minimum value, typical value, maximum value of DVDD10_USB are changed from 0.95, 1.0, 1.05 to 1.0, 1.05, 1.1 respectively.

Chapter 10 USB 2.0 Host

Figure 10-1 is simplified.

Updates in Issue 00B06 (2010-03-18)

Chapter 10 USB 2.0 Host

Figure 10-1 is simplified.

Updates in Issue 00B05 (2010-03-17)

Chapter 2 Hardware

In table 2-10, the **Type** column is added and the impedance 190 Ω of the extended resistor of the SRESREF pin is added.

Chapter 3 System

In section 3.10.4, the descriptions of SC_PERCTRL2 bit[31] and SC_PERCTRL4 bit[31] are updated.

Chapter 4 Memory Controller

In table 4-13, the corresponding pin name of NF_PAGE[1:0] is changed.

Chapter 6 Video Interface

In section 6.1.2, the time-division-multiplexing modes of port 0, port 1, port 2, and port 3 are added.

The description below Figure 6-25 is updated.

In section 6.1.7, the description of VIn_PORT_CFG bit[[7:6] is updated.

In section 6.2.1, the number of display channels supported by the VOU is changed from three to two.

In section 6.2.5, the number of surfaces that can be connected to mixer 1 and mixer 2 is changed from five to six. In addition, the driver VDC_AD of mixer 2 is changed to VDC_SD.

In section 6.2.6, the registers VO_MUX and DADACCAD are deleted.

Updates in Issue 00B04 (2010-01-19)

Chapter 4 Memory Controller

Section 4.1 is updated

Chapter 9 SATA

The registers relevant to PHY1 are deleted and the bits and descriptions of certain registers are updated.

The number of supported ports is changed from 4 to 2 and the contents relevant to port 2 and port 3 are changed.

Updates in Issue 00B03 (2009-12-23)

Chapter 1 Product Description

In section 1.1.8, the number of standard SMPTE296M and BT.1120 HD timings is changed from 1 channel to 2 channels.

Chapter 3 System

In section 3.2.3, the configuration processes of VPLL0 and VPLL1 are modified.

In section 3.10.5, the total reset value of SC_PERCLKEN is changed to 0xEFFF_CFFF.

In section 3.10.5, the value of the nf_page_size field of SC_PERCTRL23 is changed.

In Figure 3-1, PCIRSTN and relevant description are deleted

Chapter 4 Memory Interface

In section 4.1.6, the description of DDRC_EMRS01[11:9] is modified.

In section 4.3.2, the page size of 512 bytes is deleted.

In Table 4-13 and Table 4-14, the values of NF_PAGE[1:0] are changed.

In "Boot Configuration Pins" in section 4.3.5, the page size of 512 bytes is deleted.

Table 4-15 and relevant description in 00B02 are deleted.

In Table 4-17 "Common commands for operating the NAND flash memories", the last row is deleted.

In section 4.3.7, the value of the pagesize field of NFC_CON is changed.

Updates in Issue 00B02 (2009-11-30)

Complete version.

Updates in Issue 00B01(2009-06-30)

Initial version. Only chapter 1 "Product Description" and chapter 2 "Hardware" are provided.

Contents

Figures

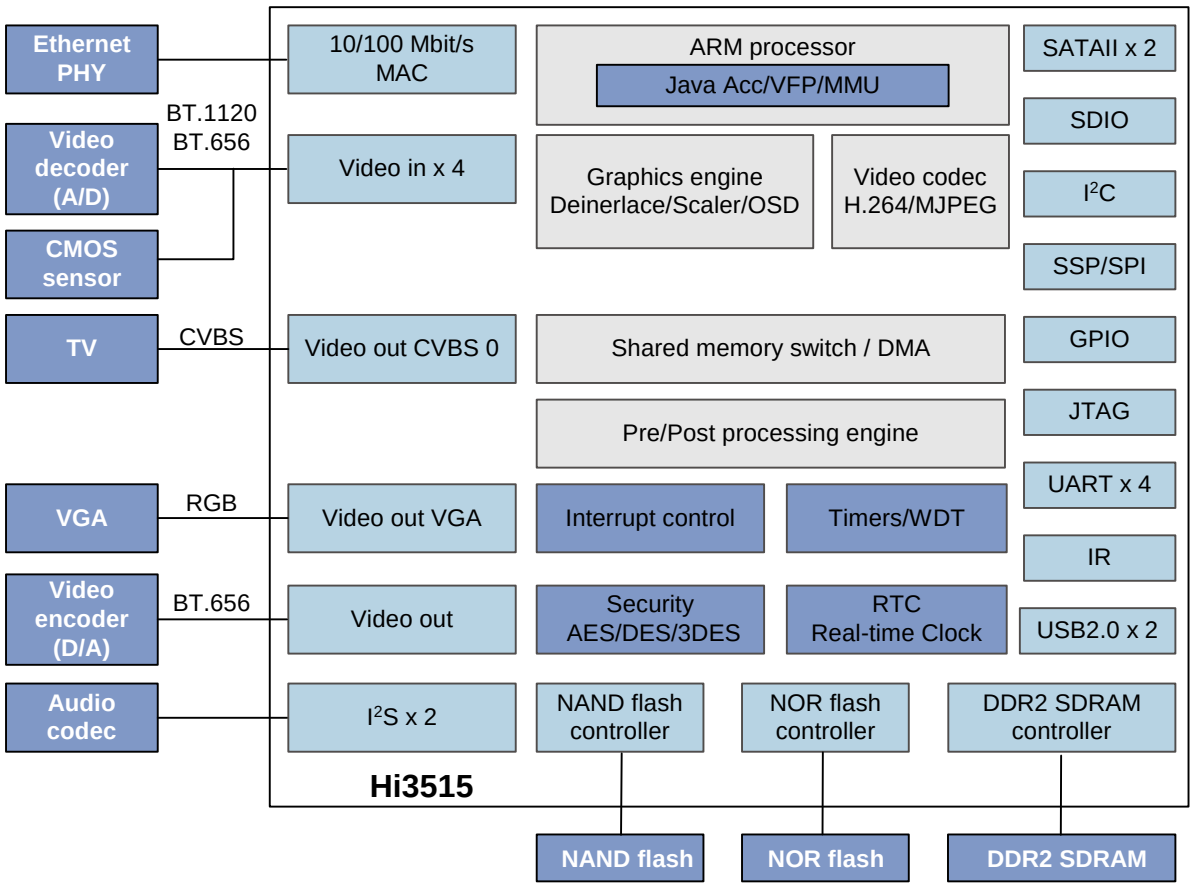
1 Product Description

1.1 Architecture

1.1.1 Overview

Figure 1-1 shows the logical block diagram of the Hi3515.

Figure 1-1 Logical block diagram of the Hi3515



1.1.2 Processor System

The Hi3515 processor is built based on the high-performance ARM926 processor. With the maximum processing frequency of 400 MHz, the ARM926 CPU provides flexible and various application services and high-performance audio/video services. The processor system consists of the following parts:

- ARM926 processor: It serves as the main control CPU to work with the hardware accelerator to encode and decode audio/video streams and schedule the system. This processor is embedded with 16 KB instruction cache, 16 KB data cache, and 2 KB instruction tightly-coupled memory (ITCM) and its operating frequency is up to 400 MHz.
- Direct memory access controller (DMAC): The DMAC can directly transfer data between a memory and a peripheral, between peripherals, and between memories.
- Interrupt system: It provides interrupt management to the entire system and supports 32 interrupt sources.
- Clock: It manages the clocks in the entire system, including the main clock of the chip and the gating clock of each module.
- Reset module: It manages the resets of the entire system and each functional module in a unified manner. To be specific, it manages and controls the power-on reset, soft reset of the system, and separate soft reset for each functional module.
- Timer: It provides two groups of dual-timers. Each group provides two independent timers.
- Watchdog: It resets the entire system when an exception occurs in the system.
- Real-time clock (RTC): It displays the time and reports alarms periodically.
- System controller: It controls the running mode of the system, monitors the running status of the system, manages key modules (such as the clock, reset module, and pin multiplexing) in the system, and configures certain functions of the peripherals.

1.1.3 Graphics Processing

The Hi3515 graphics processing module processes the video input (VI)/video output (VO) images for better display effect and strong adaptability to various scenarios. The features of the graphics processing module are as follows:

- Supports the de-interlace processing of input images
- Supports the de-interlace processing of output images for progressive display and the conversion from 60 fields to 60 frames or from 60 fields to 30 frames
- Supports color and contrast enhancement and image denoising
- Supports clip, alpha blending, raster operation (ROP), colorkey, and gamma correction
- Supports up to x16 image scaling
- Supports on-screen display (OSD) blending and video overlapping of four areas
- Supports the anti-flicker operation on output images
- Supports 2D data copying and data stuffing

1.1.4 Video Encoding and Decoding

The Hi3515 is integrated with the high-performance H.264/MJEPG/JPEG hardware codec. The features of the encoding and decoding module are as follows:

- Supports H.264 Main Profile @ Level 3 encoding/decoding
- Supports H.264 BaseLine @ Level 3 encoding/decoding
- Supports JPEG/MJPEG Baseline encoding/decoding
- Supports the maximum performance of H.264 real-time encoding/decoding, namely, 4-channel D1
- Supports the maximum resolution for H.264 encoding/decoding, namely, 1280 x 1024 @ 30 fps
- The following are supported during H.264 simultaneous encoding and decoding:
  - 240 fps CIF encoding + 240 fps QCIF encoding + 120 fps CIF decoding @ NTSC
  - 200 fps CIF encoding + 200 fps QCIF encoding + 100 fps CIF decoding @ PAL
  - 240 fps CIF encoding + 120 fps QCIF encoding + 240 fps CIF decoding @ NTSC
  - 200 fps CIF encoding + 100 fps QCIF encoding + 200 fps CIF decoding @ PAL
- Supports the encoding of dual streams complying with the H.264 or MJPEG protocol respectively
- Supports 3-megapixel encoding with 10 fps frame rate in MJPEG or JPEG format
- Controls the constant bit rate (CBR) and variable bit rate (VBR) ranging from16 kbit/s to 20 Mbit/s for H.264

1.1.5 Cipher Engine

The cipher engine uses various digital watermark encryption/decryption technologies and encryption/decryption algorithms such as the advanced encryption standard (AES) and data encryption standard (DES)/3DES. The features are as follows:

- The DES/3DES and AES algorithms comply with the FIPS46-3/FIPS 197 standards. The operating modes of DES/3DES and AES algorithms comply with the FIPS-81/NIST special 800-38a standards.
- Supports the digital watermark technology

1.1.6 Memory Controller Interface

The features of the memory controller interface are as follows:

- One double data-rate controller (DDR2) interface
  - The maximum operating rate is 200 MHz so that sufficient bandwidth is available for the entire system.
  - The data bit width is 32 bits.
  - Up to 256 MB space is supported by the synchronous dynamic random access memory (SDRAM).
- NOR flash interface
  - Supports 8-bit data width

  - Supports two chip selects (CSs). Each CS provides up to 32 MB space.
  - Supports the booting from the NOR flash

- NAND flash interface
  - Supports 8-bit data width
  - Supports the SLC and MLC and 1-bit, 4-bit, or 8-bit error correcting code (ECC) mode
  - Supports up to 8 GB capacity
  - Supports the booting from the NAND flash

1.1.7 Ethernet Interface

The 10/100 Mbit/s Ethernet interface complies with the 802.3 standard and is used for the data exchange between the external port and the main processor ARM926 in nonblock mode. The features of the Ethernet interface are as follows:

- Supports the 10/100 Mbit/s full-duplex or half-duplex mode
- Provides the media independent interface (MII)
- Supports the management data input/output (MDIO) function
- Transmits and receives flow control frames
- Supports MAC address filtering
- Supports the traffic control function
- Supports the functions of counting or debugging error packets, loss packets, ultra-short packets, ultra-long packets, unicast packets, and multicast packets in the transmit or receive process

1.1.8 Video Interface

The video interface consists of four VI interfaces, one digital VO interface, and four analog VO channels. The operating frequencies are 27 MHz, 54 MHz, and 108 MHz. The features of the video interface are as follows:

- VI interface
  - Supports 4-channel BT.656 YCrCb 4:2:2 interfaces, 8 bits, 27/54/108 MHz
  - Supports 2-channel standard SMPTE296M and BT.1120 high-definition (HD) timings
  - Supports 2-channel digital camera interfaces with the maximum resolution of 1280 x 1024 @ 30fps, 1600 x 1200 @ 20 fps, or 2048 x 1536 @ 10 fps
- VO interface
  - Supports multiple VO interfaces
  - Supports video graphics array (VGA) x 1 + composite video broadcast signal (CVBS) x 1
  - Supports typical VGA output resolutions including 1024 x 768 @ 60 fps, 1280 x 720 @ 60 fps, 1280 x 1024 @ 60 fps, 1440 x 900 @ 60 fps, and 1366 x 768 @ 60 fps
  - Supports BT.656 digital output

1.1.9 Audio Interface

The Hi3515 sonic input/output (SIO) includes two standard inter-IC sound (I^2S) interfaces that are compatible with various single- or multi-channel cascade audio codec chips. The features of the audio interface are as follows:

- Includes two I^2S interfaces. Each interface supports 16 channels of 8-bit or 16-bit audio cascade input
- Supports the sampling bit width of 8 bits, 16 bits, or 32 bits
- Supports the sampling rate of 8 kHz, 16 kHz, 32 kHz, 44.1 kHz, or 48 kHz

1.1.10 MMC/SD/SDIO Controller

The multi-media card/secure digital/secure digital input/output (MMC/SD/SDIO) controller controls the read and write operations on the SD/MMC storage cards and supports various extended devices such as Bluetooth and WiFi over the SDIO protocol. The MMC/SD/SDIO controller can control the devices that comply with the following protocols:

- SD mem-version 2.00
- SDIO-version 1.10
- MMC-version 4.2

1.1.11 USB Port

The Hi3515 integrates with two universal serial bus (USB) 2.0 host ports. The features are as follows:

- Compatible with USB 2.0
- Complies with OHCI Rev 1.0a and EHCI Rev 1.0
- Supports three transfer modes including high-speed, full-speed, and Low-speed
- Supports the low-power solution
- Supports four basic data transfer modes including control transfer, bulk transfer, isochronous transfer, and interrupt transfer
- Supports the connections to up to 127 devices through USB hubs

1.1.12 Other Peripheral Interfaces

The features of other peripheral interfaces are as follows:

- I^2C

  The inter-integrated circuit (I^2C) controller functions as standard master/slave I^2C device and complies with the Philips I^2C bus protocol. It receives/transmits data from/to the slave device of the I^2C bus.

- UART

  The universal asynchronous receiver transmitter (UART) interface refers to an asynchronous serial communications interface. By connecting it to the UART interfaces of external chips, the Hi3515 can communicate with other chips. The Hi3515 supports four UART interfaces. Three of them support basic serial data transfer. The last one supports request-to-send (RTS) and clear-to-send (CTS) hardware flow control.

- SPI

The serial peripheral interface (SPI) controller functions as a master or slave device to communicate with external devices in synchronous serial mode. The following communications protocols are supported:

- A Motorola SPI-compatible interface
- A Texas Instruments synchronous serial interface
- A National Semiconductor Microwire interface

- IR

  The infrared remoter (IR) module receives the infrared data through the infrared interface. It supports the decoding in four formats including NEC with simple repeat code, NEC with full repeat code, SONY, and TC9012. It also supports error detection on the received data and IR wake-up.

- GPIO

  The Hi3515 supports up to six groups of general-purpose input/outputs (GPIOs). Each group of GPIO provides eight programmable input/output pins. The GPIO pins are multiplexed with other service pins. The type of the pin can be configured as input or output to generate output signals or collect input signals for specific applications.

1.1.13 Hardware Specifications

- Typical power consumption of 1,500 mW in the digital video recorder (DVR) application
- Multiple-level power-saving mode
- 90 nm technology and the chip voltage of 1.0 V, 1.8 V, or 3.3 V
- 441-pin fine-pitch ball grid array (TFBGA) package, 0.8 mm ball pitch, and 19 mm x 19 mm package dimensions
- Operating temperature: –20°C to +85°C
- Surface temperature of the chip package: –20°C to +110°C
- Junction temperature: –40°C to +125°C

1.2 Application Scenarios

The Hi3515 is high-performance communications media processor based on the ARM9 processor core and video hardware accelerator engine. With the highly-integrated and programmable features, it supports various protocols such as MPEG-4 AVC/H.264 and MJPEG. Therefore, the Hi3515 is widely applied to multiple fields, such as real-time video communication and digital image surveillance.

The Hi3515 can be used in the following application scenarios:

- 4-Channel CIF DVR
- 8-Channel CIF DVR

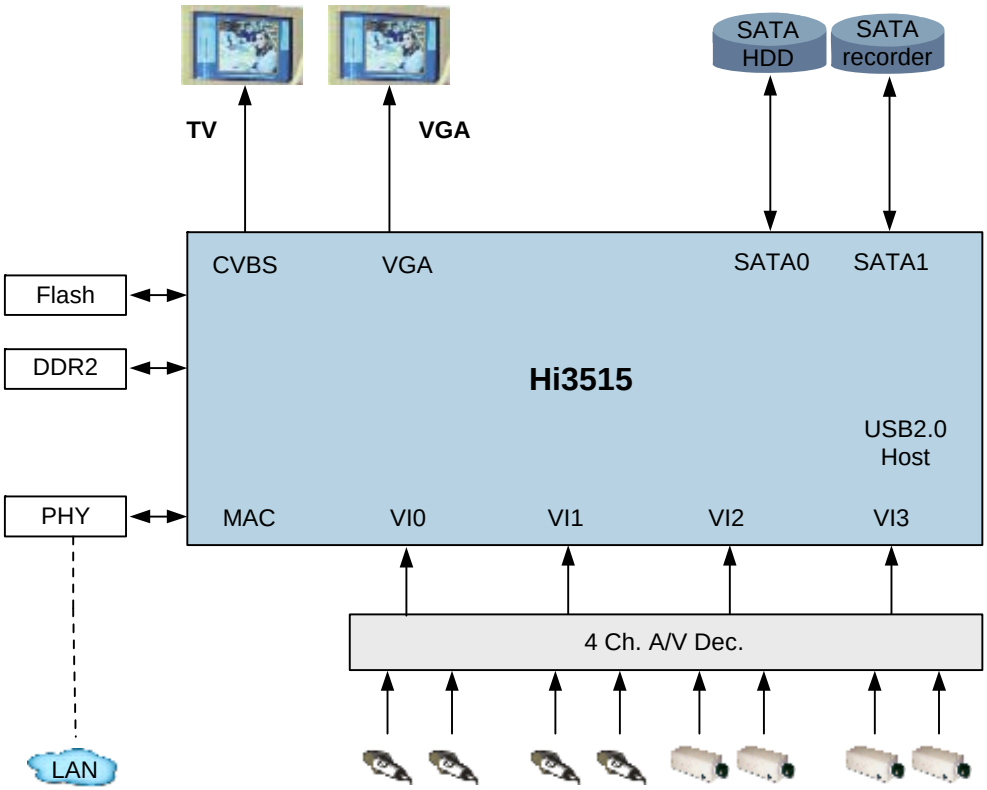
4-Channel CIF DVR

Figure 1-2 shows the block diagram of the Hi3515 in a 4-channel CIF DVR. The specifications are as follows:

- 1280 x 1024 VGA display output
- D1 spot CVBS display output

- 4-channel CIF real-time recording
- 4-channel CIF real-time network transfer
- 4-channel CIF real-time decoding replay

Figure 1-2 Block diagram of the Hi3515 in a 4-channel CIF DVR

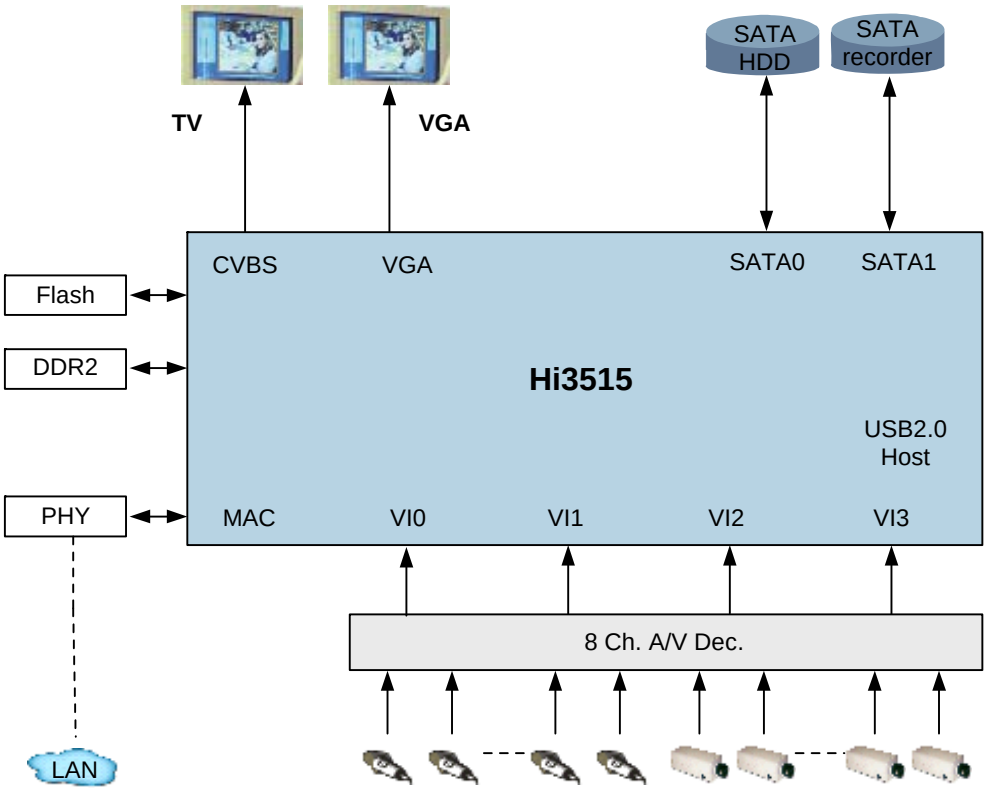


8-Channel CIF DVR

Figure 1-3 shows the block diagram of the Hi3515 in an 8-channel CIF DVR. The specifications are as follows:

- 1280 x 1024 VGA display output
- D1 spot CVBS display output
- 8-channel CIF real-time recording
- 8-channel QCIF network transfer
- 8-channel CIF real-time decoding replay

Figure 1-3 Block diagram of the Hi3515 in an 8-channel CIF DVR



Contents

Figures

Tables

2 Hardware

2.1 Pin Description

Table 2-1 describes the input/output (I/O) pin types.

Table 2-1 I/O pin types

I/O	Description
I	Input signal
I_{PD}	Input signal, internal pull-down
I_{PU}	Input signal, internal pull-up
I_S	Input signal with Schmitt trigger
I_{SPD}	Input signal with Schmitt trigger, internal pull-down
I_{SPU}	Input signal with Schmitt trigger, internal pull-up
O	Output signal
O_{OD}	Output signal, open drain (OD)
I/O	Input/output bidirectional signal
I_{PD}/O	Bidirectional signal, input pull-down
I_{PU}/O	Bidirectional signal, input pull-up
I_{SPU}/O	Bidirectional signal with Schmitt trigger, input pull-up
I_{PD}/O_{OD}	Bidirectional signal, input pull-down and output OD
I_{PU}/O_{OD}	Bidirectional signal, input pull-up and output OD
I_S/O	Bidirectional signal, input with Schmitt trigger
I_S/O_{OD}	Bidirectional signal, input with Schmitt trigger and output OD
CIN	Crystal oscillator input
COUT	Crystal oscillator output

I/O	Description
P	Power supply
G	Ground

2.1.1 Power Pins

Table 2-2 lists the power pins.

Table 2-2 Power pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
Digital-to-analog converter (DAC) power and ground					
T9	DVDD10_DAC	–	–	–	1.0 V digital power of the digital-to-analog converter (DAC)
W6	DVDD33_DAC	–	–	–	3.3 V digital power of the DAC
T8	DVSS10_DAC	–	–	–	1.0 V digital ground of the DAC
W5	DVSS33_DAC	–	–	–	3.3 V digital ground of the DAC
W4	AVDD33	–	–	–	3.3 V analog power
W3	AVDD33	–	–	–	3.3 V analog power
V5	AVDD33	–	–	–	3.3 V analog power
V4	AVDD33	–	–	–	3.3 V analog power
AC4	AVSS	–	–	–	Analog ground
AC1	AVSS	–	–	–	Analog ground
AB3	AVSS	–	–	–	Analog ground
AB2	AVSS	–	–	–	Analog ground
AA2	AVSS	–	–	–	Analog ground
Y1	AVSS	–	–	–	Analog ground
U5	AVSS	–	–	–	Analog ground

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
T5	AVSS	–	–	–	Analog ground
R8	AVSS	–	–	–	Analog ground
P8	AVSS	–	–	–	Analog ground
Phase-locked loop (PLL) power and ground					
L21	AVDD33_APLL	–	–	–	3.3 V analog power of the analog PLL (APLL)
K21	AVDD33_SPLL	–	–	–	3.3 V analog power of the software PLL (SPLL)
M19	AVDD33_VPLL	–	–	–	3.3 V analog power of the video PLL (VPLL)
L19	VDD10_APLL	–	–	–	1.0 V digital power of the APLL
K19	VDD10_SPLL	–	–	–	1.0 V digital power of the SPLL
M16	VDD10_VPLL	–	–	–	1.0 V digital power of the VPLL
L20	VSS_APLL	–	–	–	APLL ground
K20	VSS_SPLL	–	–	–	SPLL ground
M20	VSS_VPLL	–	–	–	VPLL ground
Double data-rate 2 (DDR2) power					
J16	DVDD18	–	–	–	1.8 V digital power of DDR2
H16	DVDD18	–	–	–	1.8 V digital power of DDR2
H13	DVDD18	–	–	–	1.8 V digital power of DDR2
H9	DVDD18	–	–	–	1.8 V digital power of DDR2
H8	DVDD18	–	–	–	1.8 V digital power of DDR2
G19	DVDD18	–	–	–	1.8 V digital power of DDR2
F19	DVDD18	–	–	–	1.8 V digital power of DDR2
E16	DVDD18	–	–	–	1.8 V digital power of DDR2
E15	DVDD18	–	–	–	1.8 V digital power of DDR2

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
E12	DVDD18	–	–	–	1.8 V digital power of DDR2
E11	DVDD18	–	–	–	1.8 V digital power of DDR2
E10	DVDD18	–	–	–	1.8 V digital power of DDR2
E7	DVDD18	–	–	–	1.8 V digital power of DDR2
E6	DVDD18	–	–	–	1.8 V digital power of DDR2
E18	VREF	–	–	–	0.9 V reference power of DDR2
E8	VREF	–	–	–	0.9 V reference power of DDR2
Serial advanced technology attachment (SATA) power					
R16	SPHY_VP	–	–	–	1.0 V power of the SATA
P16	SPHY_VP	–	–	–	1.0 V power of the SATA
N16	SPHY_VP	–	–	–	1.0 V power of the SATA
R19	SPHY_VPH	–	–	–	3.3 V power of the SATA
T19	SPHY_VPH	–	–	–	3.3 V power of the SATA
P19	SPHY_VPH	–	–	–	3.3 V power of the SATA
W20	SPHY_VPH	–	–	–	3.3 V power of the SATA
W19	SPHY_VPH	–	–	–	3.3 V power of the SATA
Universal serial bus (USB) power and ground					
T11	USBVDD	–	–	–	USB 2.0 HOST digital power, connected to an external 1.0 V power supply
AA11	USBVDDA33	–	–	–	USB 2.0 host analog power, connected to an external 3.3 V power supply
Y11	USBVDDA33	–	–	–	USB 2.0 HOST analog power, connected to an external 3.3 V power supply
W11	USBVDDA33	–	–	–	USB 2.0 HOST analog power, connected to an external 3.3 V power supply

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
T12	USBVSS	–	–	–	USB 2.0 HOST digital ground
AA9	USBVSSA33	–	–	–	USB 2.0 HOST analog ground
Y9	USBVSSA33	–	–	–	USB 2.0 HOST analog ground
W9	USBVSSA33	–	–	–	USB 2.0 HOST analog ground
Core power					
T16	DVDD10	–	–	–	1.0 V digital power
T15	DVDD10	–	–	–	1.0 V digital power
T14	DVDD10	–	–	–	1.0 V digital power
T13	DVDD10	–	–	–	1.0 V digital power
T10	DVDD10	–	–	–	1.0 V digital power
N8	DVDD10	–	–	–	1.0 V digital power
M8	DVDD10	–	–	–	1.0 V digital power
L16	DVDD10	–	–	–	1.0 V digital power
L8	DVDD10	–	–	–	1.0 V digital power
K16	DVDD10	–	–	–	1.0 V digital power
K8	DVDD10	–	–	–	1.0 V digital power
J8	DVDD10	–	–	–	1.0 V digital power
H15	DVDD10	–	–	–	1.0 V digital power
H14	DVDD10	–	–	–	1.0 V digital power
H11	DVDD10	–	–	–	1.0 V digital power
H10	DVDD10	–	–	–	1.0 V digital power
Other I/O power					
Y12	DVDD33	–	–	–	3.3 V digital power

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
W17	DVDD33	–	–	–	3.3 V digital power
W16	DVDD33	–	–	–	3.3 V digital power
W14	DVDD33	–	–	–	3.3 V digital power
W13	DVDD33	–	–	–	3.3 V digital power
W8	DVDD33	–	–	–	3.3 V digital power
R5	DVDD33	–	–	–	3.3 V digital power
R4	DVDD33	–	–	–	3.3 V digital power
N5	DVDD33	–	–	–	3.3 V digital power
L5	DVDD33	–	–	–	3.3 V digital power
L4	DVDD33	–	–	–	3.3 V digital power
J19	DVDD33	–	–	–	3.3 V digital power
J5	DVDD33	–	–	–	3.3 V digital power
G5	DVDD33	–	–	–	3.3 V digital power
F5	DVDD33	–	–	–	3.3 V digital power
Other digital ground					
AC23	VSS	–	–	–	Digital ground
AC11	VSS	–	–	–	Digital ground
AC9	VSS	–	–	–	Digital ground
AB11	VSS	–	–	–	Digital ground
AB9	VSS	–	–	–	Digital ground
Y18	VSS	–	–	–	Digital ground
Y15	VSS	–	–	–	Digital ground
W23	VSS	–	–	–	Digital ground
W22	VSS	–	–	–	Digital ground

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
W21	VSS	–	–	–	Digital ground
W18	VSS	–	–	–	Digital ground
W15	VSS	–	–	–	Digital ground
W12	VSS	–	–	–	Digital ground
W7	VSS	–	–	–	Digital ground
V21	VSS	–	–	–	Digital ground
U21	VSS	–	–	–	Digital ground
U20	VSS	–	–	–	Digital ground
U19	VSS	–	–	–	Digital ground
T23	VSS	–	–	–	Digital ground
T22	VSS	–	–	–	Digital ground
R21	VSS	–	–	–	Digital ground
R20	VSS	–	–	–	Digital ground
R15	VSS	–	–	–	Digital ground
R14	VSS	–	–	–	Digital ground
R13	VSS	–	–	–	Digital ground
R12	VSS	–	–	–	Digital ground
R11	VSS	–	–	–	Digital ground
R10	VSS	–	–	–	Digital ground
R9	VSS	–	–	–	Digital ground
P23	VSS	–	–	–	Digital ground
P22	VSS	–	–	–	Digital ground
P15	VSS	–	–	–	Digital ground
P14	VSS	–	–	–	Digital ground

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
P13	VSS	–	–	–	Digital ground
P12	VSS	–	–	–	Digital ground
P11	VSS	–	–	–	Digital ground
P10	VSS	–	–	–	Digital ground
P9	VSS	–	–	–	Digital ground
P5	VSS	–	–	–	Digital ground
N19	VSS	–	–	–	Digital ground
N15	VSS	–	–	–	Digital ground
N14	VSS	–	–	–	Digital ground
N13	VSS	–	–	–	Digital ground
N12	VSS	–	–	–	Digital ground
N11	VSS	–	–	–	Digital ground
N10	VSS	–	–	–	Digital ground
N9	VSS	–	–	–	Digital ground
N4	VSS	–	–	–	Digital ground
M23	VSS	–	–	–	Digital ground
M15	VSS	–	–	–	Digital ground
M14	VSS	–	–	–	Digital ground
M13	VSS	–	–	–	Digital ground
M12	VSS	–	–	–	Digital ground
M11	VSS	–	–	–	Digital ground
M10	VSS	–	–	–	Digital ground
M9	VSS	–	–	–	Digital ground
M5	VSS	–	–	–	Digital ground

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
L15	VSS	–	–	–	Digital ground
L14	VSS	–	–	–	Digital ground
L13	VSS	–	–	–	Digital ground
L12	VSS	–	–	–	Digital ground
L11	VSS	–	–	–	Digital ground
L10	VSS	–	–	–	Digital ground
L9	VSS	–	–	–	Digital ground
K22	VSS	–	–	–	Digital ground
K15	VSS	–	–	–	Digital ground
K14	VSS	–	–	–	Digital ground
K13	VSS	–	–	–	Digital ground
K12	VSS	–	–	–	Digital ground
K11	VSS	–	–	–	Digital ground
K10	VSS	–	–	–	Digital ground
K9	VSS	–	–	–	Digital ground
K5	VSS	–	–	–	Digital ground
J23	VSS	–	–	–	Digital ground
J15	VSS	–	–	–	Digital ground
J14	VSS	–	–	–	Digital ground
J13	VSS	–	–	–	Digital ground
J12	VSS	–	–	–	Digital ground
J11	VSS	–	–	–	Digital ground
J10	VSS	–	–	–	Digital ground
J9	VSS	–	–	–	Digital ground

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
J4	VSS	–	–	–	Digital ground
H20	VSS	–	–	–	Digital ground
H19	VSS	–	–	–	Digital ground
H12	VSS	–	–	–	Digital ground
H5	VSS	–	–	–	Digital ground
G21	VSS	–	–	–	Digital ground
G4	VSS	–	–	–	Digital ground
F23	VSS	–	–	–	Digital ground
E20	VSS	–	–	–	Digital ground
E19	VSS	–	–	–	Digital ground
E17	VSS	–	–	–	Digital ground
E14	VSS	–	–	–	Digital ground
E13	VSS	–	–	–	Digital ground
E9	VSS	–	–	–	Digital ground
E5	VSS	–	–	–	Digital ground
D21	VSS	–	–	–	Digital ground
D17	VSS	–	–	–	Digital ground
D15	VSS	–	–	–	Digital ground
D12	VSS	–	–	–	Digital ground
D9	VSS	–	–	–	Digital ground
D5	VSS	–	–	–	Digital ground
C22	VSS	–	–	–	Digital ground
C18	VSS	–	–	–	Digital ground
C8	VSS	–	–	–	Digital ground

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
C4	VSS	–	–	–	Digital ground
B19	VSS	–	–	–	Digital ground
B14	VSS	–	–	–	Digital ground
B7	VSS	–	–	–	Digital ground
B3	VSS	–	–	–	Digital ground
A23	VSS	–	–	–	Digital ground
A20	VSS	–	–	–	Digital ground
A14	VSS	–	–	–	Digital ground
A6	VSS	–	–	–	Digital ground
A1	VSS	–	–	–	Digital ground

2.1.2 SYS Pins

Table 2-3 lists the system (SYS) pins.

Table 2-3 SYS pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
M21	RSTN	I_{SPU}	–	–	System power-on reset input, active low.
L22	TESTMODE	I_{SPD}	–	–	Functional mode and ARM debug mode select. 0: normal mode. The ARM can be in debug mode. 1: test mode.
M22	WDGRSTN	O_T	4	–	Watchdog reset, active low and OD output.
L23	XIN24	–	2	–	24 MHz crystal oscillator clock input.
K23	XOUT24	–	–	–	24 MHz crystal oscillator clock output. If the oscillator clock is connected, this pin is floated.

2.1.3 SIO Pins

Table 2-4 lists the sonic input/output (SIO) pins.

Table 2-4 SIO pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
C1	ACKOUT	I_{SPU}/O	8	–	Programmable output clock of the SIO interface, for low-end DACs.
B2	SIO0DI	I_{PU}	–	–	Data input of SIO0. This pin can be floated when it is not in use.
D4	SIO0DO	O_T	4	–	Data output of SIO0.
C3	SIO0RCK	I_{SPU}/O	4	–	Bit stream clock on the receive side of SIO0.
D3	SIO0RFS	I_{SPU}/O	4	–	Frame sync signal on the receive side of SIO0.
C2	SIO0XCK	I_{SPU}/O	4	–	Bit stream clock on the transmit side of SIO0.
B1	SIO0XFS	I_{SPU}/O	4	–	Frame sync signal on the transmit end of SIO0.
D2	SIO1DI	I_{PU}	–	–	Data input of SIO1. This pin can be floated when it is not in use.
E4	SIO1RCK	I_{SPU}/O	4	–	Bit stream clock on the receive side of SIO1.
D1	SIO1RFS	I_{SPU}/O	4	–	Frame sync signal on the receive side of SIO1.

2.1.4 DAC Pins

Table 2-5 lists the DAC pins.

Table 2-5 DAC pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
W2	COMPDAC0	–	–	–	Compensation pin of DAC0, connected to external compensation capacitors. **Note:** It is recommended to connect a 0.01 uF ceramic capacitor and a 10 uF tantalum capacitor in parallel between this pin and the 3.3 V analog power of the DAC.
AA3	COMPDAC1	–	–	–	Compensation pin of DAC1, connected to external compensation capacitors. **Note:** It is recommended to connect a 0.01 uF ceramic capacitor and a 10 uF tantalum capacitor in parallel between this pin and the 3.3 V analog power of the DAC.
AA1	DACVGA0R	–	–	–	Red signal or 1-channel standard-definition composite video broadcast signal (CVBS) of DAC0, analog output signal.
AC3	DACVGA1B	–	–	–	Blue signal output of DAC1, analog signal.
AC2	DACVGA1G	–	–	–	Green signal of DAC1, analog signal.
AB1	DACVGA1R	–	–	–	Red signal of DAC1, analog signal.
T9	DVDD10_DAC	–	–	–	1.0 V digital power of the DAC.
W6	DVDD33_DAC	–	–	–	3.3 V digital power of the DAC.
T8	DVSS10_DAC	–	–	–	1.0 V digital ground of the DAC.
W5	DVSS33_DAC	–	–	–	3.3 V digital ground of the DAC.

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
Y2	RSETDAC0	–	–	–	Extended resistor pin of DAC0, connected to external extended resistors. **Note:** This pin is used to adjust the full-scale output current. Extended resistors must be connected between this pin and the ground.
Y3	RSETDAC1	–	–	–	Extended resistor pin of DAC1, connected to external extended resistors. **Note:** This pin is used to adjust the full-scale output current. Extended resistors must be connected between this pin and the ground.
W1	VREFINDAC0	–	–	–	Reference voltage input of DAC0. **Note:** It is recommended to connect a 0.1 uF ceramic capacitor between this pin and the analog ground of the DAC.
Y4	VREFINDAC1	–	–	–	Reference voltage input of DAC1. **Note:** It is recommended to connect a 0.1 uF ceramic capacitor between this pin and the analog ground of the DAC.

2.1.5 DDR Pins

Table 2-6 lists the DDR pins.

Table 2-6 DDR pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
A16	DDRADR0	–	–	–	Address signal of the DDR port
A11	DDRADR1	–	–	–	Address signal of the DDR port
B16	DDRADR2	–	–	–	Address signal of the DDR port
A12	DDRADR3	–	–	–	Address signal of the DDR port

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
D13	DDRADR4	–	–	–	Address signal of the DDR port
B12	DDRADR5	–	–	–	Address signal of the DDR port
C15	DDRADR6	–	–	–	Address signal of the DDR port
B11	DDRADR7	–	–	–	Address signal of the DDR port
D11	DDRADR8	–	–	–	Address signal of the DDR port
C11	DDRADR9	–	–	–	Address signal of the DDR port
C12	DDRADR10	–	–	–	Address signal of the DDR port
C13	DDRADR11	–	–	–	Address signal of the DDR port
D14	DDRADR12	–	–	–	Address signal of the DDR port
A10	DDRADR13	–	–	–	Address signal of the DDR port
D16	DDRBA0	–	–	–	Bank 0 select signal of the DDR port
C16	DDRBA1	–	–	–	Bank 1 select signal of the DDR port
C17	DDRBA2	–	–	–	Bank 2 select signal of the DDR port
A17	DDRCASN	–	–	–	Column address strobe signal of the DDR port, active low
C14	DDRCKE	–	–	–	Clock enable signal of the DDR port, active high
B15	DDRCKN0	–	–	–	Negative differential clock signal of group 0 of the DDR port
B13	DDRCKN1	–	–	–	Negative differential clock signal of group 1 of the DDR port
A15	DDRCKP0	–	–	–	Positive differential clock signal of group 0 of the DDR port
A13	DDRCKP1	–	–	–	Positive differential clock signal of group 1 of the DDR port

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
A18	DDRCSN	–	–	–	Chip select (CS) signal output from the DDR port to the DDR2 synchronous dynamic random access memory (SDRAM), active low
D19	DDRDM0	–	–	–	Byte mask signal of the DDR port, corresponding to the data bus DDRDQ[7:0]
G20	DDRDM1	–	–	–	Byte mask signal of the DDR port, corresponding to the data bus DDRDQ[15:8]
A8	DDRDM2	–	–	–	Byte mask signal of the DDR port, corresponding to the data bus DDRDQ[23:16]
A2	DDRDM3	–	–	–	Byte mask signal of the DDR port, corresponding to the data bus DDRDQ[31:24]
D18	DDRDQ0	–	–	–	Data signal of the DDR port
C20	DDRDQ1	–	–	–	Data signal of the DDR port
C19	DDRDQ2	–	–	–	Data signal of the DDR port
C21	DDRDQ3	–	–	–	Data signal of the DDR port
D20	DDRDQ4	–	–	–	Data signal of the DDR port
B20	DDRDQ5	–	–	–	Data signal of the DDR port
B22	DDRDQ6	–	–	–	Data signal of the DDR port
A22	DDRDQ7	–	–	–	Data signal of the DDR port
F20	DDRDQ8	–	–	–	Data signal of the DDR port
F22	DDRDQ9	–	–	–	Data signal of the DDR port
D22	DDRDQ10	–	–	–	Data signal of the DDR port
C23	DDRDQ11	–	–	–	Data signal of the DDR port
F21	DDRDQ12	–	–	–	Data signal of the DDR port
E21	DDRDQ13	–	–	–	Data signal of the DDR port
D23	DDRDQ14	–	–	–	Data signal of the DDR port

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
B23	DDRDQ15	–	–	–	Data signal of the DDR port
B8	DDRDQ16	–	–	–	Data signal of the DDR port
C10	DDRDQ17	–	–	–	Data signal of the DDR port
D8	DDRDQ18	–	–	–	Data signal of the DDR port
D10	DDRDQ19	–	–	–	Data signal of the DDR port
B10	DDRDQ20	–	–	–	Data signal of the DDR port
A7	DDRDQ21	–	–	–	Data signal of the DDR port
C9	DDRDQ22	–	–	–	Data signal of the DDR port
C7	DDRDQ23	–	–	–	Data signal of the DDR port
C5	DDRDQ24	–	–	–	Data signal of the DDR port
A3	DDRDQ25	–	–	–	Data signal of the DDR port
C6	DDRDQ26	–	–	–	Data signal of the DDR port
D7	DDRDQ27	–	–	–	Data signal of the DDR port
B6	DDRDQ28	–	–	–	Data signal of the DDR port
A4	DDRDQ29	–	–	–	Data signal of the DDR port
B4	DDRDQ30	–	–	–	Data signal of the DDR port
D6	DDRDQ31	–	–	–	Data signal of the DDR port
A21	DDRDQSN0	–	–	–	Data strobe (DQS) negative differential signal of the DDR port, corresponding to the data bus DDRDQ[7:0]
E23	DDRDQSN1	–	–	–	DQS negative differential signal of the DDR port, corresponding to the data bus DDRDQ[15:8]
B9	DDRDQSN2	–	–	–	DQS negative differential signal of the DDR port, corresponding to the data bus DDRDQ[23:16]
B5	DDRDQSN3	–	–	–	DQS negative differential signal of the DDR port, corresponding to the data bus DDRDQ[31:24]

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
B21	DDRDQSP0	–	–	–	DQS positive differential signal of the DDR port, corresponding to the data bus DDRDQ[7:0]
E22	DDRDQSP1	–	–	–	DQS positive differential signal of the DDR port, corresponding to the data bus DDRDQ[15:8]
A9	DDRDQSP2	–	–	–	DQS positive differential signal of the DDR port, corresponding to the data bus DDRDQ[23:16]
A5	DDRDQSP3	–	–	–	DQS positive differential signal of the DDR port, corresponding to the data bus DDRDQ[31:24]
B18	DDRODT	–	–	–	On-die termination (ODT) signal of the DDR port
B17	DDRRASN	–	–	–	Row address strobe signal of the DDR port, active low
A19	DDRWEN	–	–	–	Write enable signal of the DDR port, active low

2.1.6 EBI Pins

Table 2-7 lists the external bus interface (EBI) pins.

Table 2-7 EBI pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
AB14	EBIADR0	O	8	–	EBI address bus ADR0
AA14	EBIADR1	O	8	–	EBI address bus ADR1
Y14	EBIADR2	O	8	–	EBI address bus ADR2
AC15	EBIADR3	O	8	–	EBI address bus ADR3
AB15	EBIADR4	O	8	–	EBI address bus ADR4
AA15	EBIADR5	O	8	–	EBI address bus ADR5
AC16	EBIADR6	O	8	–	EBI address bus ADR6
AB16	EBIADR7	O	8	–	EBI address bus ADR7

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
AA16	EBIADR8	O	8	–	EBI address bus ADR8
Y16	EBIADR9	O	8	–	EBI address bus ADR9
AC17	EBIADR10	O	8	–	EBI address bus ADR10
AB17	EBIADR11	O	8	–	EBI address bus ADR11
AA17	EBIADR12	O	8	–	EBI address bus ADR12
AC18	EBIADR13	O	8	–	EBI address bus ADR13
AB18	EBIADR14	O	8	–	EBI address bus ADR14
AC19	EBIADR15	I/O	8	–	EBI address bus ADR15
AB19	EBIADR16	I/O	8	–	EBI address bus ADR16
AC20	EBIADR17	I/O	8	–	EBI address bus ADR17
AB20	EBIADR18	I/O	8	–	EBI address bus ADR18
AC21	EBIADR19	I/O	8	–	EBI address bus ADR19
AB21	EBIADR20	I/O	8	–	EBI address bus ADR20
AC22	EBIADR21	O	8	–	EBI address bus ADR21
AB22	EBIADR22	I/O	8	–	EBI address bus ADR22
AB23	EBIADR23	O	8	–	EBI address bus ADR23
AA18	EBIADR24	I/O	8	–	EBI address bus ADR24
AC12	EBIDQ0	I/O	8	–	EBI data bus DAT0
AB12	EBIDQ1	I/O	8	–	EBI data bus DAT1
AA12	EBIDQ2	I/O	8	–	EBI data bus DAT2
AC13	EBIDQ3	I/O	8	–	EBI data bus DAT3
AB13	EBIDQ4	I/O	8	–	EBI data bus DAT4
AA13	EBIDQ5	I/O	8	–	EBI data bus DAT5
Y13	EBIDQ6	I/O	8	–	EBI data bus DAT6

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
AC14	EBIDQ7	I/O	8	–	EBI data bus DAT7
AA21	EBIRDYN	I_{SPU}/O	4	–	Static memory interface (SMI) input ready indicator signal, active low. This pin can be floated when it is not in use.
AA20	EBIWEN	O	8	–	EBI write signal, active low
AA22	NFALE	O	4	–	Address latch signal of the NAND flash
Y20	NFCLE	O	4	–	Command latch signal of the NAND flash
Y22	NFCS0N	O	4	–	NAND flash CS 0, active low. The system can boot from this CS.
Y23	NFCS1N	I/O	4	–	NAND flash CS 1, active low
Y21	NFOEN	O	8	–	Read enable signal of the NAND flash, active low
AA23	NFRB	I_{PU}/O	4	–	Status indicator signal of the NAND flash. 0: busy 1: idle If two external NAND flashes are connected, they must be wire-ANDed together and then connected to this pin.
AA19	SMICS0N	O	4	–	SMI CS signal 0, active low (default) or active high. The system can boot from this CS.
Y19	SMICS1N	I/O	4	–	SMI CS signal 1, active low (default) or active high.
Y17	SMIOEN	O	8	–	Read enable signal of the SMI interface, active low

2.1.7 ETH Pins

Table 2-8 lists the Ethernet (ETH) pins.

Table 2-8 ETH pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
AB7	ECOL	I_{PU}/O	4	–	ETH collision indicator signal
AA7	ECRS	I_{PU}/O	4	–	ETH carrier sense signal
Y7	ERXCK	I_{PU}	–	–	ETH receive clock input
AC8	ERXD0	I_{PU}	–	–	ETH receive data DAT0
AB8	ERXD1	I_{PU}	–	–	ETH receive data DAT1
AA8	ERXD2	I_{PU}	–	–	ETH receive data DAT2
Y8	ERXD3	I_{PU}	–	–	ETH receive data DAT3
AC6	ERXDV	I_{PU}	–	–	ETH receive data valid
AA6	ETXCK	I_{PU}	–	–	ETH transmit clock input
AB5	ETXD0	O	4	–	ETH transmit data DAT0
AC5	ETXD1	O	4	–	ETH transmit data DAT1
Y6	ETXD2	O	4	–	ETH transmit data DAT2
AB6	ETXD3	O	4	–	ETH transmit data DAT3
AC7	ETXEN	O	4	–	ETH transmit enable signal
Y5	MDCK	O	4	–	Clock output of the management data input/output (MDIO) interface
AA5	MDIO	I_{PU}/O	4	–	Input/output signal of the MDIO interface

2.1.8 I²C Pins

Table 2-9 lists the inter-integrated circuit (I^2C) pins.

Table 2-9 I^2C pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
J21	SCL	I_{SPU}/O	4	–	I^2C bus clock, OD output. This pin must have an external pull-up resister on the printed circuit board (PCB).
J20	SDA	I_{SPU}/O	4	–	I^2C bus data/address, OD output. This pin must have an external pull-up resister on the PCB.

2.1.9 SATA Pins

Table 2-10 lists the SATA pins.

Table 2-10 SATA pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
R22	SREFCKM	I	–	–	SATA negative differential clock input
R23	SREFCKP	I	–	–	SATA positive differential clock input
P20	SRESREF	I/O	–	–	SATA extended resistor pin, connected to 190 Ω external extended resistors
V23	SRXP0	I	–	–	Positive differential data input of SATA port 0
U22	SRXM1	I	–	–	Negative differential data input of SATA port 1
V22	SRXM0	I	–	–	Negative differential data input of SATA port 0
U23	SRXP1	I	–	–	Positive differential data input of SATA port 1
V19	STXM0	O	–	–	Negative differential data output of SATA port 0
T21	STXM1	O	–	–	Negative differential data output of SATA port 1
V20	STXP0	O	–	–	Positive differential data output of SATA port 0

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
T20	STXP1	O	–	–	Positive differential data output of SATA port 1

2.1.10 JTAG Pins

Table 2-11 lists the joint test action group (JTAG) pins.

Table 2-11 JTAG pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
N20	TCK	I_{SPD}	–	–	JTAG clock input
N23	TDI	I_{SPU}	–	–	JTAG data input
N22	TDO	O_T	4	–	JTAG data output.
N21	TMS	I_{SPU}	–	–	JTAG mode select input
P21	TRSTN	I_{SPD}	–	–	JTAG reset input, active low

2.1.11 UART Pin

Table 2-12 lists the universal asynchronous receiver transmitter (UART) pins.

Table 2-12 UART pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
G23	UCTSN1	I_{PU}	–	–	CTSN signal of UART1. This pin can be floated when it is not in use.
H22	URTSN1	O_T	4	–	RTSN signal of UART1.
H23	URXD0	I_{PU}	–	–	UART0 data receive
H21	URXD1	I_{PU}	–	–	UART1 data receive
J22	UTXD0	O_T	4	–	UART0 data transmit
G22	UTXD1	O_T	4	–	UART1 data transmit

2.1.12 USB Pins

Table 2-13 lists the USB pins.

Table 2-13 USB pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
AB10	USBDM0	I/O	–	–	D- differential data bus of USB 2.0 HOST port 0, analog signal
Y10	USBDM1	I/O	–	–	D- differential data bus of the USB 2.0 HOST port 1, analog signal
AC10	USBDP0	I/O	–	–	D+ differential data bus of USB 2.0 HOST port 0, analog signal
AA10	USBDP1	I/O	–	–	D+ differential data bus of USB 2.0 HOST port 1, analog signal
W10	USBREXT	I/O	–	–	USB 2.0 HOST external resistor connected end

2.1.13 VO Pins

Table 2-14 lists the video output (VO) pins.

Table 2-14 VO pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
AA4	VGAHS	O	8	–	Video graphics array (VGA) horizontal sync signal
AB4	VGAVS	O	8	–	VGA vertical sync signal
F3	VOCK	I/O	8	–	Standard-definition (SD) image output clock
E3	VODAT0	I_{SPU}/O	8	–	SD image output data DAT0
E2	VODAT1	I_{SPU}/O	8	–	SD image output data DAT1
E1	VODAT2	I_{SPU}/O	8	–	SD image output data DAT2
F4	VODAT3	I_{SPU}/O	8	–	SD image output data DAT3
F2	VODAT4	I_{SPU}/O	8	–	SD image output data DAT4

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
F1	VODAT5	I_{SPU}/O	8	–	SD image output data DAT5
G3	VODAT6	I/O	8	–	SD image output data DAT6
G2	VODAT7	I/O	8	–	SD image output data DAT7

2.1.14 VI Pins

Table 2-15 lists the video input (VI) pins.

Table 2-15 VI pins

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
H2	VI0CK	I_{PU}	–	–	Clock of the VI0 interface
H4	VI0DAT0	I_{PU}	–	–	Data DAT0 of the VI0 interface
H3	VI0DAT1	I_{PU}	–	–	Data DAT1 of the VI0 interface
J3	VI0DAT2	I_{PU}	–	–	Data DAT2 of the VI0 interface
J2	VI0DAT3	I_{PU}	–	–	Data DAT3 of the VI0 interface
J1	VI0DAT4	I_{PU}	–	–	Data DAT4 of the VI0 interface
K4	VI0DAT5	I_{PU}	–	–	Data DAT5 of the VI0 interface
K3	VI0DAT6	I_{PU}	–	–	Data DAT6 of the VI0 interface
K2	VI0DAT7	I_{PU}	–	–	Data DAT7 of the VI0 interface
G1	VI0HS	I_{PU}/O	4	–	Horizontal sync signal of the VI0 interface. This pin can be floated when it is not in use.
H1	VI0VS	I_{PU}/O	4	–	Vertical sync signal of the VI0 interface. This pin can be floated when it is not in use.
M4	VI1CK	I_{PU}	–	–	Clock of the VI1 interface

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
K1	VI1DAT0	I_{PU}/O	4	–	Data DAT0 of the VI1 interface
L3	VI1DAT1	I_{PU}/O	4	–	Data DAT1 of the VI1 interface
L2	VI1DAT2	I_{PU}/O	4	–	Data DAT2 of the VI1 interface
L1	VI1DAT3	I_{PU}/O	4	–	Data DAT3 of the VI1 interface
M3	VI1DAT4	I_{PU}/O	4	–	Data DAT4 of the VI1 interface
M2	VI1DAT5	I_{PU}/O	4	–	Data DAT5 of the VI1 interface
M1	VI1DAT6	I_{PU}/O	4	–	Data DAT6 of the VI1 interface
N1	VI1DAT7	I_{PU}/O	4	–	Data DAT7 of the VI1 interface
R3	VI2CK	I_{PU}	–	–	Clock of the VI2 interface
P4	VI2DAT0	I_{PU}/O	4	–	Data DAT0 of the VI2 interface
P3	VI2DAT1	I_{PU}/O	4	–	Data DAT1 of the VI2 interface
P2	VI2DAT2	I_{PU}/O	4	–	Data DAT2 of the VI2 interface
P1	VI2DAT3	I_{PU}/O	4	–	Data DAT3 of the VI2 interface
R2	VI2DAT4	I_{PU}/O	4	–	Data DAT4 of the VI2 interface
R1	VI2DAT5	I_{PU}/O	4	–	Data DAT5 of the VI2 interface
T2	VI2DAT6	I_{PU}/O	4	–	Data DAT6 of the VI2 interface
T1	VI2DAT7	I_{PU}/O	4	–	Data DAT7 of the VI2 interface
N3	VI2HS	I_{PU}/O	4	–	Horizontal sync signal of the VI2 interface. This pin can be floated when it is not in use.

Pin	Pin Name	Type	Drive (mA)	Voltage (V)	Description
N2	VI2VS	I_{PU}/O	4	–	Vertical sync signal of the VI2 interface. This pin can be floated when it is not in use.
U3	VI3CK	I_{PU}	–	–	Clock of the VI3 interface
T4	VI3DAT0	I_{PU}/O	4	–	Data DAT0 of the VI3 interface
T3	VI3DAT1	I_{PU}/O	4	–	Data DAT1 of the VI3 interface
U1	VI3DAT2	I_{PU}/O	4	–	Data DAT2 of the VI3 interface
U2	VI3DAT3	I_{PU}/O	4	–	Data DAT3 of the VI3 interface
U4	VI3DAT4	I_{PU}/O	4	–	Data DAT4 of the VI3 interface
V1	VI3DAT5	I_{PU}/O	4	–	Data DAT5 of the VI3 interface
V2	VI3DAT6	I_{PU}/O	4	–	Data DAT6 of the VI3 interface
V3	VI3DAT7	I_{PU}/O	4	–	Data DAT7 of the VI3 interface

2.2 Description of Software Multiplexing Pins

Software multiplexing refers to a function that the CPU controls the pin multiplexing by configuring registers.

VI

Table 2-16 lists the software multiplexing pins of the VI interface.

Table 2-16 Software multiplexing pins of the VI interface

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1	Multiplexing Signal 2
G1	VI0HS	reg0	URXD2	GPIO1_3
H1	VI0VS	reg1	UTXD2	GPIO1_4

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1	Multiplexing Signal 2
K1	VI1DAT0	reg2	GPIO3_0	–
L3	VI1DAT1	reg3	GPIO3_1	–
L2	VI1DAT2	reg4	GPIO3_2	–
L1	VI1DAT3	reg5	GPIO3_3	–
M3	VI1DAT4	reg6	GPIO3_4	–
M2	VI1DAT5	reg7	GPIO3_5	–
M1	VI1DAT6	reg8	GPIO3_6	–
N1	VI1DAT7	reg9	GPIO3_7	–
N3	VI2HS	reg10	URXD3	GPIO1_5
N2	VI2VS	reg11	UTXD3	GPIO1_6
P4	VI2DAT0	reg12	GPIO4_0	–
P3	VI2DAT1	reg13	GPIO4_1	–
P2	VI2DAT2	reg14	GPIO4_2	–
P1	VI2DAT3	reg15	GPIO4_3	–
R2	VI2DAT4	reg16	GPIO4_4	–
R1	VI2DAT5	reg17	GPIO4_5	–
T2	VI2DAT6	reg18	GPIO4_6	–
T1	VI2DAT7	reg19	GPIO4_7	–
T4	VI3DAT0	reg20	GPIO5_0	–
T3	VI3DAT1	reg21	GPIO5_1	–
U1	VI3DAT2	reg22	GPIO5_2	–
U2	VI3DAT3	reg23	GPIO5_3	–
U4	VI3DAT4	reg24	GPIO5_4	–
V1	VI3DAT5	reg25	GPIO5_5	–

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1	Multiplexing Signal 2
V2	VI3DAT6	reg26	GPIO5_6	–
V3	VI3DAT7	reg27	GPIO5_7	–

Table 2-17 describes the software multiplexing pins of the VI interface.

Table 2-17 Description of the software multiplexing pins of the VI interface

Signal	Direction	Description
GPIO1_3	I/O	General purpose input/output (GPIO)
GPIO1_4	I/O	GPIO
GPIO1_5	I/O	GPIO
GPIO1_6	I/O	GPIO
GPIO3_0	I/O	GPIO
GPIO3_1	I/O	GPIO
GPIO3_2	I/O	GPIO
GPIO3_3	I/O	GPIO
GPIO3_4	I/O	GPIO
GPIO3_5	I/O	GPIO
GPIO3_6	I/O	GPIO
GPIO3_7	I/O	GPIO
GPIO4_0	I/O	GPIO
GPIO4_1	I/O	GPIO
GPIO4_2	I/O	GPIO
GPIO4_3	I/O	GPIO
GPIO4_4	I/O	GPIO
GPIO4_5	I/O	GPIO
GPIO4_6	I/O	GPIO
GPIO4_7	I/O	GPIO
GPIO5_0	I/O	GPIO
GPIO5_1	I/O	GPIO

Signal	Direction	Description
GPIO5_2	I/O	GPIO
GPIO5_3	I/O	GPIO
GPIO5_4	I/O	GPIO
GPIO5_5	I/O	GPIO
GPIO5_6	I/O	GPIO
GPIO5_7	I/O	GPIO
URXD2	I	UART2 data receive
URXD3	I	UART3 data receive
UTXD2	O	UART2 data transmit
UTXD3	O	UART0 data transmit

VO

Table 2-18 lists the software multiplexing pins of the VO interface.

Table 2-18 Software multiplexing pins of the VO interface

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1	Multiplexing Signal 2	Multiplexing Signal 3
F3	VOCK	reg28	GPIO1_7	SDIOCK	SPICK
E3	VODAT0	reg29	GPIO2_0	SDIOCMD	SPIDI
E2	VODAT1	reg30	GPIO2_1	SDIODAT0	SPIDO
E1	VODAT2	reg31	GPIO2_2	SDIODAT1	SPICSN0
F4	VODAT3	reg32	GPIO2_3	SDIODAT2	SPICSN1
F2	VODAT4	reg33	GPIO2_4	SDIODAT3	–
F1	VODAT5	reg34	GPIO2_5	SDIODETC	–
G3	VODAT6	reg35	SATALEDN0	–	–
G2	VODAT7	reg36	SATALEDN1	–	–

Table 2-19 describes the software multiplexing pins of the VO interface.

Table 2-19 Description of the software multiplexing pins of the VO interface

Signal	Direction	Description
GPIO1_7	I/O	GPIO
GPIO2_0	I/O	GPIO
GPIO2_1	I/O	GPIO
GPIO2_2	I/O	GPIO
GPIO2_3	I/O	GPIO
GPIO2_4	I/O	GPIO
GPIO2_5	I/O	GPIO
SATALEDN0	I/O	SATA light emitting diode (LED) 0
SATALEDN1	I/O	SATA LED 1
SDIOCK	O	Secure digital input/output (SDIO)/Multi-media card (MMC) clock
SDIOCMD	I/O	SDIO/MMC command
SDIODAT0	I/O	SDIO/MMC data DAT0
SDIODAT1	I/O	SDIO/MMC data DAT1
SDIODAT2	I/O	SDIO/MMC data DAT2
SDIODAT3	I/O	SDIO/MMC data DAT3
SDIODETC	I	SDIO/MMC card detection signal
SPICK	I/O	Serial peripheral interface (SPI) clock signal
SPICSN0	I/O	SPI CS 0, active low
SPICSN1	I/O	SPI CS 1, active low
SPIDI	I	SPI input data. This pin can be floated when it is not in use.
SPIDO	O	SPI output data

I²C

Table 2-20 lists the software multiplexing pins of the I²C interface.

Table 2-20 Software multiplexing pins of the I²C interface

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1
J20	SDA	reg37	GPIO0_0

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1
J21	SCL	reg38	GPIO0_1

Table 2-21 describes the software multiplexing pins of the I^2C interface.

Table 2-21 Description of the software multiplexing pins of the I^2C interface

Signal	Direction	Description
GPIO0_0	I/O	GPIO
GPIO0_1	I/O	GPIO

SIO

Table 2-22 lists the software multiplexing pins of the SIO interface.

Table 2-22 Software multiplexing pins of the SIO interface

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1
B1	SIO0XFS	reg39	GPIO0_2
C2	SIO0XCK	reg40	GPIO0_3
C1	ACKOUT	reg41	GPIO0_4

Table 2-23 describes the software multiplexing pins of the SIO interface.

Table 2-23 Description of the software multiplexing pins of the SIO interface

Signal	Direction	Description
GPIO0_2	I/O	GPIO
GPIO0_3	I/O	GPIO
GPIO0_4	I/O	GPIO

EBI

Table 2-24 lists the software multiplexing pins of the EBI interface.

Table 2-24 Software multiplexing pins of the EBI interface

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1	Multiplexing Signal 2
Y19	SMICS1N	reg42	GPIO0_5	-
Y23	NFCS1N	reg43	GPIO0_6	-
AA23	NFRB	reg44	GPIO0_7	-
AA21	EBIRDYN	reg45	IRRCV	GPIO1_0

Table 2-25 describes the software multiplexing pins of the EBI interface.

Table 2-25 Description of the software multiplexing pins of the EBI interface

Signal	Direction	Description
GPIO0_5	I/O	GPIO
GPIO0_6	I/O	GPIO
GPIO0_7	I/O	GPIO
GPIO1_0	I/O	GPIO
IRRCV	I	Infrared remote receive

ETH

Table 2-26 lists the software multiplexing pins of the ETH interface.

Table 2-26 Software multiplexing pins of the ETH interface

Pin	Pad Signal	Multiplexing Control Register	Multiplexing Signal 1
AB7	ECOL	reg46	GPIO1_1
AA7	ECRS	reg47	GPIO1_2

Table 2-27 describes the software multiplexing pins of the ETH interface.

Table 2-27 Description of the software multiplexing pins of the ETH interface

Signal	Direction	Description
GPIO1_1	I/O	GPIO

Signal	Direction	Description
GPIO1_2	I/O	GPIO

2.3 Description of Hardware Multiplexing Pins

EBI Pin Multiplexing

NOTE

In Table 2-28, power_on==0'b1 indicates the time of canceling the hardware reset of the Hi3515. When designing the following pins on the PCB, you must connect a pull-up resistor or pull-down resistor to select required modes.

Table 2-28 describes the multiplexing of the EBI pins.

Table 2-28 Hardware multiplexing of the EBI pins

Pin	Pad Signal	Multiplexing Signal 1 (power_on == 0'b1)
AC19	EBIADR15	NFECC0
AB19	EBIADR16	NFECC1
AC20	EBIADR17	NFNUM0
AB20	EBIADR18	NFNUM1
AC21	EBIADR19	NFPAGE0
AB21	EBIADR20	NFPAGE1
AB22	EBIADR22	FUNSEL
AA18	EBIADR24	BOOTSEL

Table 2-29 lists the multiplexed pins of the EBI pins.

Table 2-29 Multiplexed pins of the EBI pins

Signal	Direction	Description
NFECC0	I	ECC type when the system boots from the NAND flash
NFECC1	I	ECC type when the system boots from the NAND flash
NFNUM0	I	Number of addresses when the system boots from the NAND flash

Signal	Direction	Description
NFNUM1	I	Number of addresses when the system boots from the NAND flash
NFPAGE0	I	Page size when the system boots from the NAND flash
NFPAGE1	I	Page size when the system boots from the NAND flash
FUNSEL	I	Functional mode select. 0: ARM debug 1: SATA debug
BOOTSEL	I	Boot mode select. 00: boot from the NOR flash 00: boot from the NAND flash

2.4 Summary of the IO Config (Pin Multiplexing Control) Registers

Table 2-30 lists the pin multiplexing control registers.

Table 2-30 Summary of the pin multiplexing control registers (base address: 0x200F_0000)

Offset Address	Register	Description	Page
0x0000	reg0	Multiplexing control register of the VI0HS pin	2-38
0x0004	reg1	Multiplexing control register of the VI0VS pin	2-38
0x0008	reg2	Multiplexing control register of the VI1DAT0 pin	2-39
0x000C	reg3	Multiplexing control register of the VI1DAT1 pin	2-39
0x0010	reg4	Multiplexing control register of the VI1DAT2 pin	2-40
0x0014	reg5	Multiplexing control register of the VI1DAT3 pin	2-40
0x0018	reg6	Multiplexing control register of the VI1DAT4 pin	2-41
0x001C	reg7	Multiplexing control register of the VI1DAT5 pin	2-41

Offset Address	Register	Description	Page
0x0020	reg8	Multiplexing control register of the VI1DAT6 pin	2-41
0x0024	reg9	Multiplexing control register of the VI1DAT7 pin	2-42
0x0028	reg10	Multiplexing control register of the VI2HS pin	2-42
0x002C	reg11	Multiplexing control register of the VI2VS pin	2-43
0x0030	reg12	Multiplexing control register of the VI2DAT0 pin	2-43
0x0034	reg13	Multiplexing control register of the VI2DAT1 pin	2-44
0x0038	reg14	Multiplexing control register of the VI2DAT2 pin	2-44
0x003C	reg15	Multiplexing control register of the VI2DAT3 pin	2-45
0x0040	reg16	Multiplexing control register of the VI2DAT4 pin	2-45
0x0044	reg17	Multiplexing control register of the VI2DAT5 pin	2-46
0x0048	reg18	Multiplexing control register of the VI2DAT6 pin	2-46
0x004C	reg19	Multiplexing control register of the VI2DAT7 pin	2-47
0x0050	reg20	Multiplexing control register of the VI3DAT0 pin	2-47
0x0054	reg21	Multiplexing control register of the VI3DAT1 pin	2-48
0x0058	reg22	Multiplexing control register of the VI3DAT2 pin	2-48
0x005C	reg23	Multiplexing control register of the VI3DAT3 pin	2-49
0x0060	reg24	Multiplexing control register of the VI3DAT4 pin	2-49
0x0064	reg25	Multiplexing control register of the VI3DAT5 pin	2-50
0x0068	reg26	Multiplexing control register of the VI3DAT6 pin	2-50

Offset Address	Register	Description	Page
0x006C	reg27	Multiplexing control register of the VI3DAT7 pin	2-51
0x0070	reg28	Multiplexing control register of the VOCK pin	2-51
0x0074	reg29	Multiplexing control register of the VODAT0 pin	2-52
0x0078	reg30	Multiplexing control register of the VODAT1 pin	2-52
0x007C	reg31	Multiplexing control register of the VODAT2 pin	2-53
0x0080	reg32	Multiplexing control register of the VODAT3 pin	2-53
0x0084	reg33	Multiplexing control register of the VODAT4 pin	2-54
0x0088	reg34	Multiplexing control register of the VODAT5 pin	2-54
0x008C	reg35	Multiplexing control register of the VODAT6 pin	2-55
0x0090	reg36	Multiplexing control register of the VODAT7 pin	2-55
0x0094	reg37	Multiplexing control register of the serial data (SDA) pin	2-56
0x0098	reg38	Multiplexing control register of the serial clock (SCL) pin	2-56
0x009C	reg39	Multiplexing control register of the SIO0XFS pin	2-57
0x00A0	reg40	Multiplexing control register of the SIO0XCK pin	2-57
0x00A4	reg41	Multiplexing control register of the ACKOUT pin	2-58
0x00A8	reg42	Multiplexing control register of the SMICS1N pin	2-58
0x00AC	reg43	Multiplexing control register of the NFCS1N pin	2-59
0x00B0	reg44	Multiplexing control register of the NFRB pin	2-59
0x00B4	reg45	Multiplexing control register of the EBIRDYN pin	2-60

Offset Address	Register	Description	Page
0x00B8	reg46	Multiplexing control register of the ECOL pin	2-60
0x00BC	reg47	Multiplexing control register of the ECRS pin	2-61

reg0

Multiplexing control register of the VI0HS pin.

Offset Address	Register Name	Total Reset Value
0x0000	reg0	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg0	Multiplexing of the VI0HS pin. 00: VI0HS 01: URXD2 10: GPIO1_3 Others: reserved

reg1

Multiplexing control register of the VI0VS pin.

Offset Address	Register Name	Total Reset Value
0x0004	reg1	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg1	

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg1	Multiplexing of the VI0VS pin. 00: VI0VS 01: UTXD2 10: GPIO1_4 Others: reserved

reg2

Multiplexing control register of the VI1DAT0 pin.

Offset Address	Register Name	Total Reset Value
0x0008	reg2	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg2
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg2	Multiplexing of the VI1DAT0 pin. 0: VI1DAT0 1: GPIO3_0

reg3

Multiplexing control register of the VI1DAT1 pin.

Offset Address	Register Name	Total Reset Value
0x000C	reg3	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg3
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg3	Multiplexing of the VI1DAT1 pin.

			0: VI1DAT1 1: GPIO3_1

reg4

Multiplexing control register of the VI1DAT2 pin.

Offset Address	Register Name	Total Reset Value
0x0010	reg4	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg4
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg4	Multiplexing of the VI1DAT2 pin. 0: VI1DAT2 1: GPIO3_2

reg5

Multiplexing control register of the VI1DAT3 pin.

Offset Address	Register Name	Total Reset Value
0x0014	reg5	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg5
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg5	Multiplexing of the VI1DAT3 pin. 0: VI1DAT3 1: GPIO3_3

reg6

Multiplexing control register of the VI1DAT4 pin.

Offset Address	Register Name	Total Reset Value
0x0018	reg6	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg6
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg6	Multiplexing of the VI1DAT4 pin. 0: VI1DAT4 1: GPIO3_4

reg7

Multiplexing control register of the VI1DAT5 pin.

Offset Address	Register Name	Total Reset Value
0x001C	reg7	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg7
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg7	Multiplexing of the VI1DAT5 pin. 0: VI1DAT5 1: GPIO3_5

reg8

Multiplexing control register of the VI1DAT6 pin.

Offset Address	Register Name	Total Reset Value
0x0020	reg8	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg8
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg8	Multiplexing of the VI1DAT6 pin. 0: VI1DAT6 1: GPIO3_6

reg9

Multiplexing control register of the VI1DAT7 pin.

Offset Address	Register Name	Total Reset Value
0x0024	reg9	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg9
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg9	Multiplexing of the VI1DAT7 pin. 0: VI1DAT7 1: GPIO3_7

reg10

Multiplexing control register of the VI2HS pin.

Offset Address	Register Name	Total Reset Value
0x0028	reg10	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg10	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg10	Multiplexing of the VI2HS pin. 00: VI2HS 01: URXD3 10: GPIO1_5 Others: reserved

reg11

Multiplexing control register of the VI2VS pin.

Offset Address	Register Name	Total Reset Value
0x002C	reg11	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg11	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg11	Multiplexing of the VI2VS pin. 00: VI2VS 01: UTXD3 10: GPIO1_6 Others: reserved

reg12

Multiplexing control register of the VI2DAT0 pin.

Offset Address	Register Name	Total Reset Value
0x0030	reg12	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg12
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg12	Multiplexing of the VI2DAT0 pin. 0: VI2DAT0 1: GPIO4_0

reg13

Multiplexing control register of the VI2DAT1 pin.

Offset Address	Register Name	Total Reset Value
0x0034	reg13	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg13
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg13	Multiplexing of the VI2DAT1 pin. 0: VI2DAT1 1: GPIO4_1

reg14

Multiplexing control register of the VI2DAT2 pin.

Offset Address	Register Name	Total Reset Value
0x0038	reg14	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg14
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg14	Multiplexing of the VI2DAT2 pin. 0: VI2DAT2 1: GPIO4_2

reg15

Multiplexing control register of the VI2DAT3 pin.

Offset Address	Register Name	Total Reset Value
0x003C	reg15	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg15
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg15	Multiplexing of the VI2DAT3 pin. 0: VI2DAT3 1: GPIO4_3

reg16

Multiplexing control register of the VI2DAT4 pin.

Offset Address	Register Name	Total Reset Value
0x0040	reg16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg16	Multiplexing of the VI2DAT4 pin. 0: VI2DAT4 1: GPIO4_4

reg17

Multiplexing control register of the VI2DAT5 pin.

Offset Address	Register Name	Total Reset Value
0x0044	reg17	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg17
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg17	Multiplexing of the VI2DAT5 pin. 0: VI2DAT5 1: GPIO4_5

reg18

Multiplexing control register of the VI2DAT6 pin.

Offset Address	Register Name	Total Reset Value
0x0048	reg18	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg18
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg18	Multiplexing of the VI2DAT6 pin. 0: VI2DAT6 1: GPIO4_6

reg19

Multiplexing control register of the VI2DAT7 pin.

Offset Address	Register Name	Total Reset Value
0x004C	reg19	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg19
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg19	Multiplexing of the VI2DAT7 pin. 0: VI2DAT7 1: GPIO4_7

reg20

Multiplexing control register of the VI3DAT0 pin.

Offset Address	Register Name	Total Reset Value
0x0050	reg20	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg20
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg20	Multiplexing of the VI3DAT0 pin. 0: VI3DAT0 1: GPIO5_0

reg21

Multiplexing control register of the VI3DAT1 pin.

Offset Address	Register Name	Total Reset Value
0x0054	reg21	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg21
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg21	Multiplexing of the VI3DAT1 pin. 0: VI3DAT1 1: GPIO5_1

reg22

Multiplexing control register of the VI3DAT2 pin.

Offset Address	Register Name	Total Reset Value
0x0058	reg22	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg22
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg22	Multiplexing of the VI3DAT2 pin. 0: VI3DAT2 1: GPIO5_2

reg23

Multiplexing control register of the VI3DAT3 pin.

Offset Address	Register Name	Total Reset Value
0x005C	reg23	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg23
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg23	Multiplexing of the VI3DAT3 pin. 0: VI3DAT3 1: GPIO5_3

reg24

Multiplexing control register of the VI3DAT4 pin.

Offset Address	Register Name	Total Reset Value
0x0060	reg24	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg24
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg24	Multiplexing of the VI3DAT4 pin. 0: VI3DAT4 1: GPIO5_4

reg25

Multiplexing control register of the VI3DAT5 pin.

Offset Address	Register Name	Total Reset Value
0x0064	reg25	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg25
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg25	Multiplexing of the VI3DAT5 pin. 0: VI3DAT5 1: GPIO5_5

reg26

Multiplexing control register of the VI3DAT6 pin.

Offset Address	Register Name	Total Reset Value
0x0068	reg26	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg26
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg26	Multiplexing of the VI3DAT6 pin. 0: VI3DAT6 1: GPIO5_6

reg27

Multiplexing control register of the VI3DAT7 pin.

Offset Address	Register Name	Total Reset Value
0x006C	reg27	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg27
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg27	Multiplexing of the VI3DAT7 pin. 0: VI3DAT7 1: GPIO5_7

reg28

Multiplexing control register of the VOCK pin.

Offset Address	Register Name	Total Reset Value
0x0070	reg28	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg28	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg28	Multiplexing of the VOCK pin. 00: GPIO1_7 01: VOCK 10: SDIOCK 11: SPICK

reg29

Multiplexing control register of the VODAT0 pin.

Offset Address	Register Name	Total Reset Value
0x0074	reg29	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg29	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg29	Multiplexing of the VODAT0 pin. 00: GPIO2_0 01: VODAT0 10: SDIOCMD 11: SPIDI

reg30

Multiplexing control register of the VODAT1 pin.

Offset Address	Register Name	Total Reset Value
0x0078	reg30	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg30	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg30	Multiplexing of the VODAT1 pin. 00: GPIO2_1 01: VODAT1 10: SDIODAT0 11: SPIDO

reg31

Multiplexing control register of the VODAT2 pin.

Offset Address	Register Name	Total Reset Value
0x007C	reg31	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg31	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg31	Multiplexing of the VODAT2 pin. 00: GPIO2_2 01: VODAT2 10: SDIODAT1 11: SPICSN0

reg32

Multiplexing control register of the VODAT3 pin.

Offset Address	Register Name	Total Reset Value
0x0080	reg32	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg32	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg32	Multiplexing of the VODAT3 pin. 00: GPIO2_3 01: VODAT3 10: SDIODAT2 11: SPICSN1

reg33

Multiplexing control register of the VODAT4 pin.

Offset Address	Register Name	Total Reset Value
0x0084	reg33	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg33	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg33	Multiplexing of the VODAT4 pin. 00: GPIO2_4 01: VODAT4 10: SDIODAT3 Others: reserved

reg34

Multiplexing control register of the VODAT5 pin.

Offset Address	Register Name	Total Reset Value
0x0088	reg34	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																														reg34	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg34	Multiplexing of the VODAT5 pin. 00: GPIO2_5 01: VODAT5 10: SDIODETC Others: reserved

reg35

Multiplexing control register of the VODAT6 pin.

Offset Address	Register Name	Total Reset Value
0x008C	reg35	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg35
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg35	Multiplexing of the VODAT6 pin. 0: SATALEDN0 1: VODAT6

reg36

Multiplexing control register of the VODAT7 pin.

Offset Address	Register Name	Total Reset Value
0x0090	reg36	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg36
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg36	Multiplexing of the VODAT7 pin. 0: SATALEDN1 1: VODAT7

reg37

Multiplexing control register of the SDA pin.

Offset Address	Register Name	Total Reset Value
0x0094	reg37	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg37
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg37	Multiplexing of the SDA pin. 0: SDA 1: GPIO0_0

reg38

Multiplexing control register of the SCL pin.

Offset Address	Register Name	Total Reset Value
0x0098	reg38	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg38
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg38	Multiplexing of the SCL pin. 0: SCL 1: GPIO0_1

reg39

Multiplexing control register of the SIO0XFS pin.

Offset Address	Register Name	Total Reset Value
0x009C	reg39	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg39
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg39	Multiplexing of the SIO0XFS pin. 0: SIO0XFS 1: GPIO0_2

reg40

Multiplexing control register of the SIO0XCK pin.

Offset Address	Register Name	Total Reset Value
0x00A0	reg40	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg40
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg40	Multiplexing of the SIO0XCK pin. 0: SIO0XCK 1: GPIO0_3

reg41

Multiplexing control register of the ACKOUT pin.

Offset Address	Register Name	Total Reset Value
0x00A4	reg41	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg41
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg41	Multiplexing of the ACKOUT pin. 0: GPIO0_4 1: ACKOUT

reg42

Multiplexing control register of the SMICS1N pin.

Offset Address	Register Name	Total Reset Value
0x00A8	reg42	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg42
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg42	Multiplexing of the SMICS1N pin. 0: GPIO0_5 1: SMICS1N

reg43

Multiplexing control register of the NFCS1N pin.

Offset Address	Register Name	Total Reset Value
0x00AC	reg43	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg43
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg43	Multiplexing of the NFCS1N pin. 0: GPIO0_6 1: NFCS1N

reg44

Multiplexing control register of the NFRB pin.

Offset Address	Register Name	Total Reset Value
0x00B0	reg44	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg44
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg44	Multiplexing of the NFRB pin. 0: NFRB 1: GPIO0_7

reg45

Multiplexing control register of the EBIRDYN pin.

Offset Address	Register Name	Total Reset Value
0x00B4	reg45	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg45
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[1:0]	RW	reg45	Multiplexing of the EBIRDYN pin. 00: EBIRDYN 01: IRRCV 10: GPIO1_0 Others: reserved

reg46

Multiplexing control register of the ECOL pin.

Offset Address	Register Name	Total Reset Value
0x00B8	reg46	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg46
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg46	Multiplexing of the ECOL pin. 0: ECOL 1: GPIO1_1

reg47

Multiplexing control register of the ECRS pin.

Offset Address	Register Name	Total Reset Value
0x00BC	reg47	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															reg47
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[0]	RW	reg47	Multiplexing of the ECRS pin. 0: ECRS 1: GPIO1_2

2.5 Recommended Power-on and Power-off Sequences

It is recommended to power on the VDD33, VDD18, and VDD10 in sequence, and power off the VDD10, VDD18, and VDD33 in sequence.

2.6 External Interrupts

For details about external interrupts, see section 3.4 "Interrupt System."

2.7 Electrical Specifications

2.7.1 DC/AC Parameters

Table 2-31, Table 2-32, and Table 2-33 list the direct current (DC) and alternating current (AC) parameters.

Table 2-31 DC parameters (DVDD33 = 3.3 V)

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
V_{IH}	Input high voltage	2.0	–	5.5	V	–
V_{IL}	Input low voltage	–0.3	–	0.8	V	–
V_{OH}	Output high voltage	2.4	–	–	–	–
V_{OL}	Output low voltage	–	–	0.4	V	–
I_L	Input leakage current	–	–	±10	μA	–
I_{OZ}	Tri-state output leakage current	–	–	±10	μA	–
R_{PU}	Pull-up resistor	63	92	142	kΩ	–
R_{PD}	Pull-down resistor	57	91	159	kΩ	–

Table 2-32 DC parameters (DVDD18 = 1.8 V)

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
Vref	Reference voltage	0.85	0.9	0.95	V	0.5 x VDD18
$V_{IH\ (dc)}$	Input high voltage	Vref + 0.125	–	DVDD18 + 0.3	V	–
$V_{IL(dc)}$	Input low voltage	–0.3	–	Vref - 0.125	V	–
$I_{OH(dc)}$	Output high current	|–13.4|	–	–	mA	Voh(dc) = 1.42 V
$I_{OL(dc)}$	Output low current	13.4	–	–	mA	Vol(dc) = 0.28 V
V_{TT}	Valid termination voltage	Vref - 0.04	Vref	Vref + 0.04	V	–

Table 2-33 AC parameters (DVDD18 = 1.8 V)

Symbol	Parameter	Min	Typ	Max	Unit	Remarks
$V_{IH(ac)}$	Input high voltage	Vref + 0.25	–	–	V	–
$V_{IL(ac)}$	Input low voltage	–	–	Vref - 0.25	V	–
$V_{OH(ac)}$	Output high voltage	V_{TTmax} + 0.603	–	–	V	I_{OH} = -13.4 mA
$V_{OL(ac)}$	Output low voltage	–	–	V_{TTmin} - 0.603	V	I_{OL} = 13.4 mA
SLEW	Minimum input skew	1.0	–	–	V/ns	-

2.7.2 Recommended Operating Conditions

Table 2-34 lists the recommended operating conditions.

Table 2-34 Recommended operating conditions

Symbol	Parameter	Min	Typ	Max	Unit
TOPT	Ambient temperature	–20	–	85	°C
TS	Surface temperature of the chip package	–20	–	110	°C
TJ	Junction temperature	–40	–	125	°C
DVDD10	Core power	1.0	1.05	1.15	V
DVDD33	I/O power	3.135	3.3	3.6	V
DVDD18	DDR I/O power	1.7	1.8	1.9	V
VREF	DDR reference power (0.5 x VDD18)	0.85	0.9	0.95	V
AVDD33_PLL	PLL analog power	3.135	3.3	3.6	V
DVDD10_PLL	PLL digital power	1.0	1.05	1.15	V
AVDD33_USB	USB analog power	3.135	3.3	3.465	V
DVDD10_USB	USB digital power	1.0	1.05	1.1	V
AVDD33_VDAC	VDAC analog power	3.135	3.3	3.6	V
DVDD10_DAC	VDAC digital power	1.0	1.05	1.15	V
DVDD33_DAC	VDAC digital power	3.0	3.3	3.6	V
SPHY_VP	SATA core power	1.0	1.05	1.15	V

Symbol	Parameter	Min	Typ	Max	Unit
SPHY_VPH	SATA IO power	3.0	3.3	3.6	V

2.8 PCB Routing Recommendations

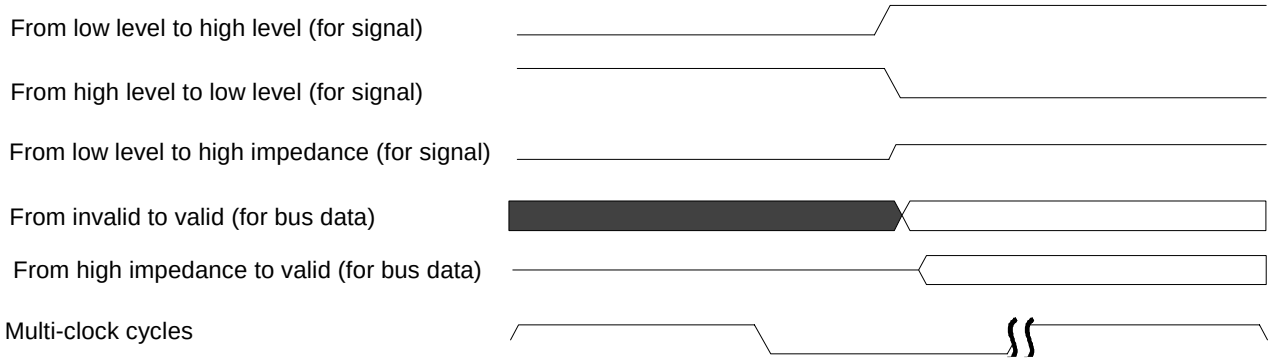
For details about the PCB routing recommendations, see the *Hi3515 Hardware Design User Guide*.

2.9 Timing Specifications

2.9.1 Primitives of Timing Diagrams

Figure 2-1 shows the primitives of the timing diagrams in this document.

Figure 2-1 Primitives of timing diagrams



2.9.2 Timings of the DDR2 Interface

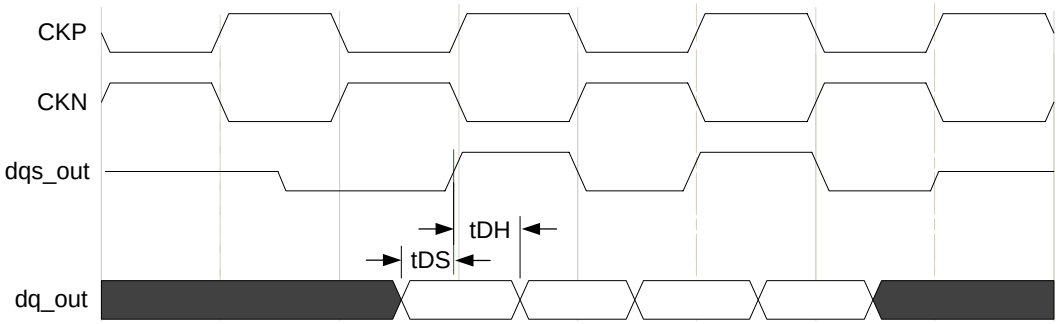
2.9.2.1 Write Timings

Write Timing of dqs_out Relative to dq_out

In the write timing of dqs_out relative to dq_out, the parameters tDS and tDH need to be checked. In DDR2-400, tDS is 0.150 ns and tDH is 0.275 ns.

Figure 2-2 shows the timing of dqs_out relative to dq_out and the DDR clocks CKP and CKN.

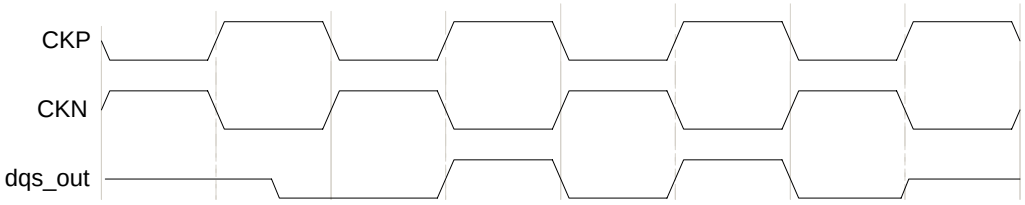
Figure 2-2 Write timing of dqs_out relative to dq_out



Write Timing of dqs_out Relative to CK

In the write timing of dqs_out relative to CK, the parameters tDSS and tDSH need to be checked. Their values are tCK x 0.2. Figure 2-3 shows the write timing.

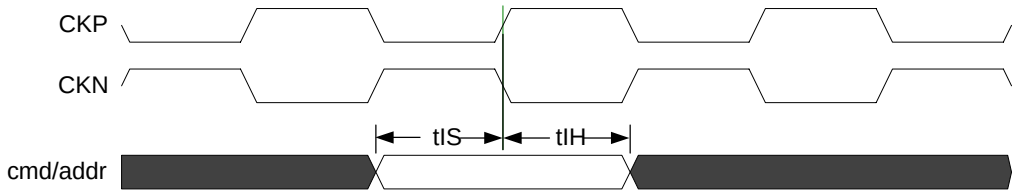
Figure 2-3 Write timing of dqs_out relative to CK



Write Timing of cmd/addr Relative to CK

Figure 2-4 shows the write timing of cmd/addr relative to CK.

Figure 2-4 Write timing of cmd/addr relative to CK



2.9.2.2 Read Timings

Read Timing of cmd/addr Relative to CK

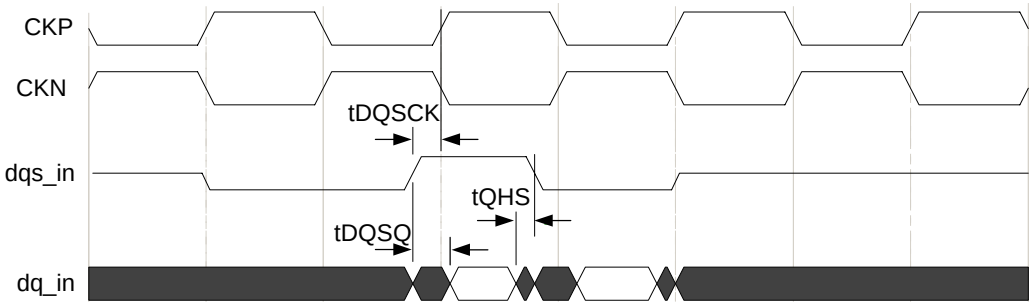
For details, see the description of Write Timing of cmd/addr Relative to CK in section 2.9.2.1 .

Read Timing of dqs_in Relative to dq_in

For the DDR2 SDRAM output timing, DQS and CK are in phase in the ideal case. Actually, there is a skew of tDQSCK and the skew is 0.500 ns. tDQSQ is the jitter of the last valid DQ relative to DQS and its value is 0.350 ns; tQHS is the jitter of the first valid DQ relative to DQS and its value is 0.450 ns.

Figure 2-5 shows the DDR2 SDRAM output timing.

Figure 2-5 DDR2 SDRAM output timing



2.9.2.3 Timing Parameters

All the timings described in this document are at the DDR physical layer entity sublayer (PHY) side. The Hi3515 parameters are based on the timing parameters of DDR2-400. For details, see Table 2-35 to Table 2-37.

Table 2-35 Clock parameters

Parameter	Typ	Unit
Memory clock frequency	200.00	MHz
PLL clock jitter	0.200	ns
PLL duty cycle	45.000	%
CLK Skew	0.200	ns

Table 2-36 Memory device parameters

Parameter	Typ	Unit
Memory type	DDR2-400	–
tDSS: DQS falling edge-CK setup time	0.2	Tck
tDSH: DQS falling edge-CK hold time	0.2	Tck
tDS: DQ/DM-DQS setup time	0.150	ns
tDH: DQ/DM-DQS hold time	0.275	ns
DQS-DQ skew, tDQSQ	0.350	ns
Data hold skew factor, tQHS	0.450	ns

Parameter	Typ	Unit
Adr/Cmd setup time, tIS	0.350	ns
Adr/Cmd hold time, tIH	0.475	ns
tDQSCK: DQS output access time from CKP/CKN (min)	–0.500	ns
tDQSCK: DQS output access time from CKP/CKN (max)	0.500	ns

Table 2-37 Package/Board parameters

Parameter	Typ	Unit
Package trace skew (max/min)	0.050	ns
Board trace length mismatch (for Dqs/Dq)	0.050	ns
Board trace length mismatch (for others)	0.100	ns
Board and package uncertainty (output)	0.250	ns
Board and package uncertainty (input)	0.250	ns

2.9.3 Timings of the ETH Interface

The Hi3515 provides a standard media independent interface (MII) that complies with the MII interface timing standard. This interface is used to connect to the PHY chip.

Timings of the MII Interface

Figure 2-6 shows the 100 Mbit/s receive timing of the MII interface.

Figure 2-6 100 Mbit/s receive timing of the MII interface

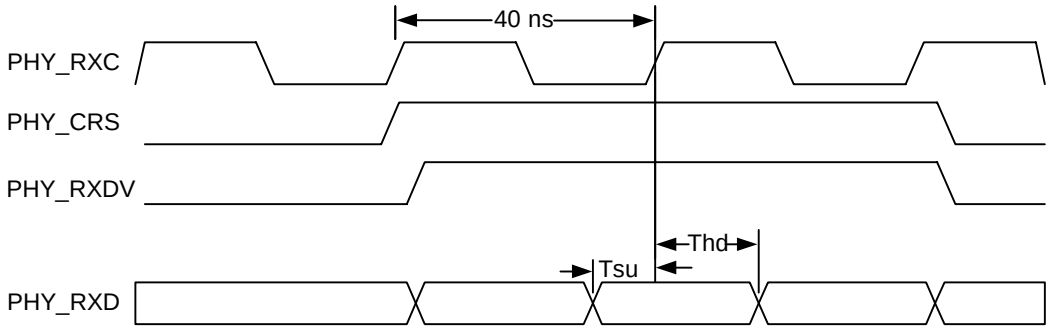
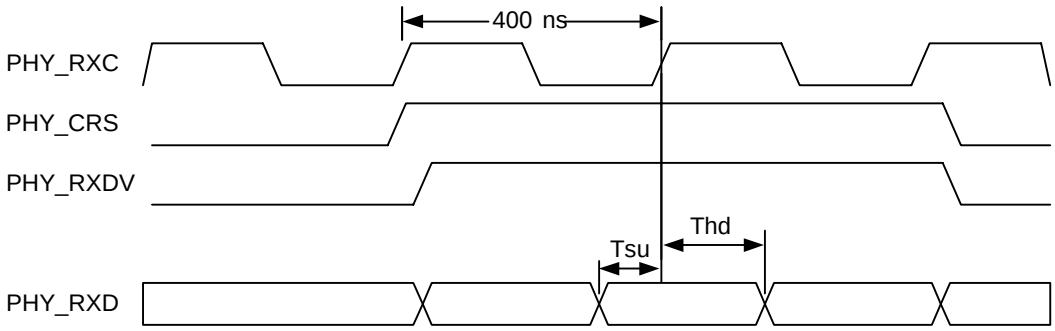


Figure 2-7 shows the 10 Mbit/s receive timing of the MII interface.

Figure 2-7 10 Mbit/s receive timing of the MII interface



Transmit Timings of the MII Interface

Figure 2-8 shows the 100 Mbit/s transmit timing of the MII interface.

Figure 2-8 100 Mbit/s transmit timing of the MII interface

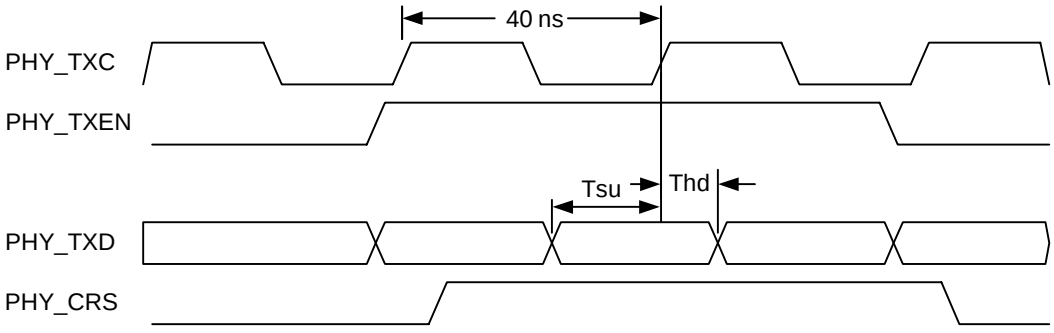


Figure 2-9 shows the 10 Mbit/s transmit timing of the MII interface.

Figure 2-9 10 Mbit/s transmit timing of the MII interface

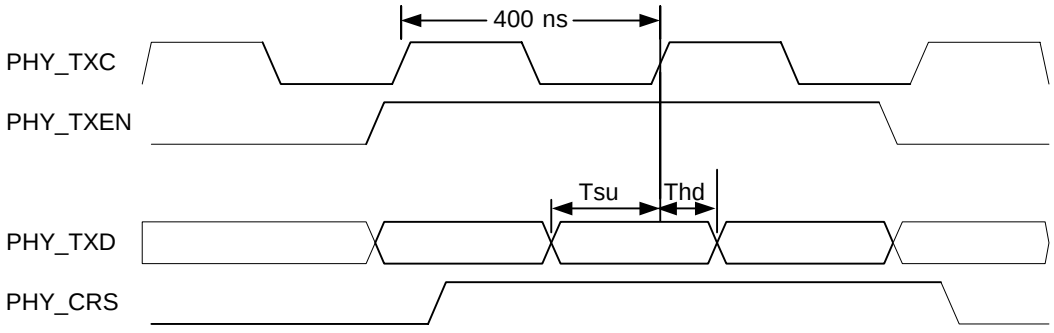


Figure 2-10 shows the timing parameter diagram of the MII interface.

Figure 2-10 Timing parameter diagram of the MII interface

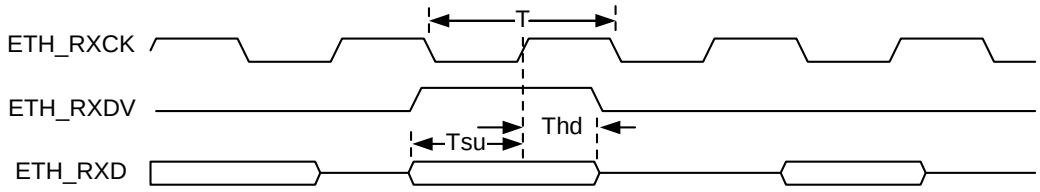


Figure 2-11 shows the transmit timing parameter diagram of the MII interface.

Figure 2-11 Transmit timing parameter diagram of the MII interface

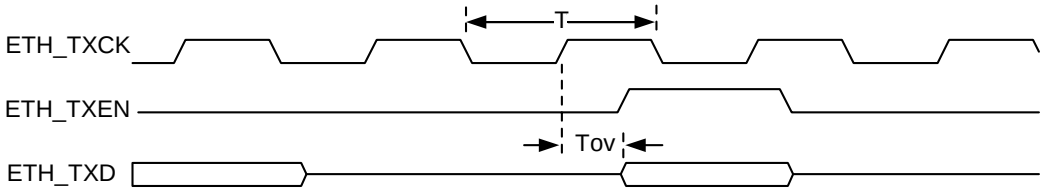


Table 2-38 lists the timing parameters of the MII interface.

Table 2-38 Timing parameters of the MII interface

Parameter	Symbol	Signal	Min	Max	Unit
MII clock cycle	T	ETH_RXCK, ETH_TXCK	400 (10Mbit/s)	400	ns
			40 (10Mbit/s)	40	ns
MII signal setup time	Tsu (RX)	ETH_RXER, ETH_RXDV, ETH_RXD[3:0]	6	–	ns
MII signal hold time	Thd (RX)	ETH_RXER, ETH_RXDV, ETH_RXD[3:0]	2	–	ns
MII output signal delay	Tov (TX)	ETH_TXD[1:0] , ETH_TXEN	2	8	ns

Timings of the MDIO Interface

Figure 2-12 shows the read timing of the MDIO interface.

Figure 2-12 Read timing of the MDIO interface

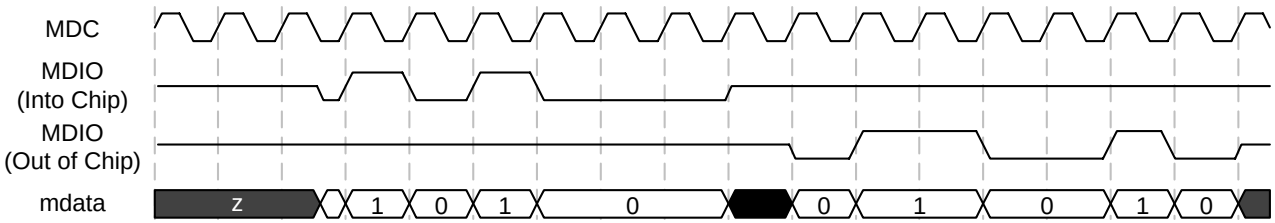


Figure 2-13 shows the write timing of the MDIO interface.

Figure 2-13 Write timing of the MDIO interface

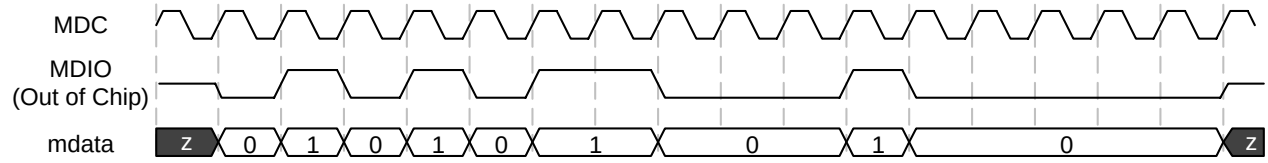


Figure 2-14 shows the timing parameter diagram of the MDIO interface.

Figure 2-14 Timing parameter diagram of the MDIO interface

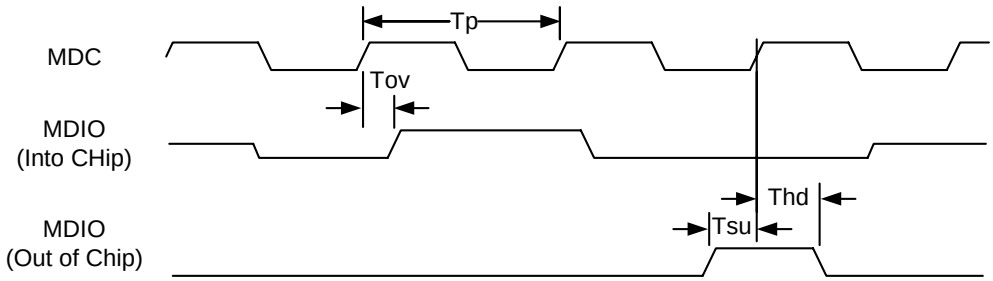


Table 2-39 lists the timing parameters of the MDIO interface.

Table 2-39 Timing parameters of the MDIO interface

Parameter	Symbol	Signal	Min	Max	Unit
MDIO receive data delay	Tov	MDIO	166	20833	ns
MDIO clock cycle	Tp	MDC	333	41667	ns
MDIO transmit data setup time	Tsu	MDIO	10	–	ns
MDIO transmit data hold time	Thd	MDIO	10	–	ns

NOTE

The clock cycle Tp can be changed by adjusting the frequency of the MDC (MDIO_RWCTRL[frq_dv]). You can divide the 150 MHz ETH working clock by 100, 50, or other values. Tov is related to Tp and it is about Tmdc/2.

2.9.4 Timings of the VI Interface

Figure 2-15 shows the timings of the VI interface.

Figure 2-15 Timings of the VI interface

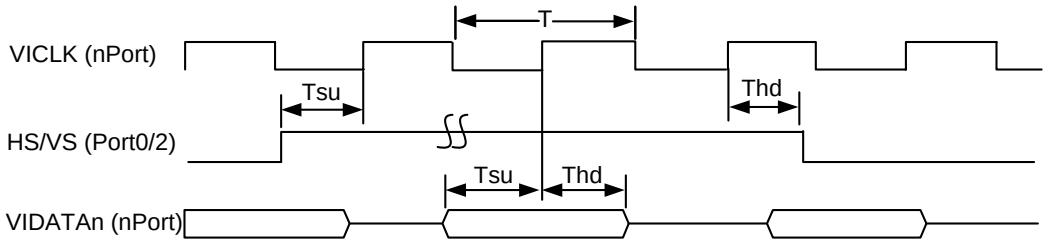


Table 2-40 lists the timing parameters of the VI interface.

Table 2-40 Timing parameters of the VI interface

Parameter	Symbol	Min	Typ	Max	Unit
VICLK clock cycle (the VIU clock depends on its supported protocol)	T	–	–	–	ns
Input signal setup time	Tsu	2.93	–	–	ns
Input signal hold time	Thd	2	–	–	ns

2.9.5 Timings of the VO Interface

Figure 2-16 shows the timing of the VO interface.

Figure 2-16 Timings of the VO interface

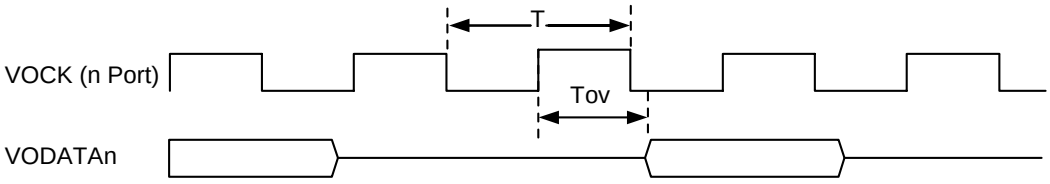


Table 2-41 lists the timing parameters of the VO interface.

Table 2-41 Timing parameters of the VO interface

Parameter	Symbol	Min	Typ	Max	Unit
VOCK clock cycle	T	–	37.03	–	ns
Output signal delay	Tov	T/2 - △T	T/2 + 1	T/2 + 1 + △T	ns

NOTE

The value of △T depends on its supported protocol. For example, △T is 3 T when the BT.656 protocol is supported; △T is 0.11 T when the BT.1120 protocol is supported.

2.9.6 Timings of the I²C Interface

Figure 2-17 shows the transfer timings of the I²C interface.

Figure 2-17 Transfer timings of the I²C interface

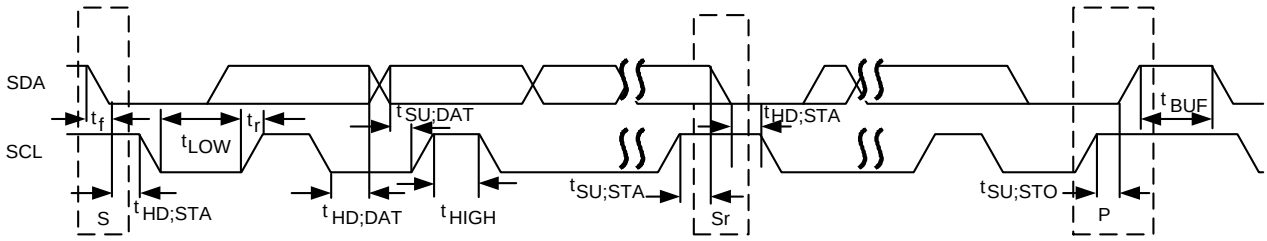


Table 2-42 lists the timing parameters of the I²C interface.

Table 2-42 Timing parameters of the I²C interface

Parameter	Symbol	Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	
SCL clock frequency	f_{SCL}	0	100	0	400	kHz
Start hold time	$t_{HD;STA}$	4.0	–	0.6	–	μs
SCL low-level cycle	t_{LOW}	4.7	–	1.3	–	μs
SCL high-level cycle	t_{HIGH}	4.0	–	0.6	–	μs
Start setup time	$t_{SU;STA}$	4.7	–	0.6	–	μs
Data hold time	$t_{HD;DAT}$	0	3.45	0	0.9	μs
Data setup time	$t_{SU;DAT}$	250	–	100	–	ns
SDA and SCL rising time	t_r	–	1000	20 + 0.1C_b	300	ns
SDA and SCL falling time	t_f	–	300	20 + 0.1C_b	300	ns

Parameter	Symbol	Standard Mode		Fast Mode		Unit
		Min	Max	Min	Max	
End setup time	$t_{SU;STO}$	4.0	–	0.6	–	μs
Bus release time from start to end	t_{BUF}	4.7	–	1.3	–	μs
Bus load	C_b	–	400	–	400	pF
Low-level noise tolerance	V_{nL}	$0.1V_{DD}$	–	$0.1V_{DD}$	–	V
High-level noise tolerance	V_{nH}	$0.2V_{DD}$	–	$0.2V_{DD}$	–	V

2.9.7 Timings of the MMC Interface

Figure 2-18 shows the timings of the MMC interface.

Figure 2-18 Timings of the MMC interface

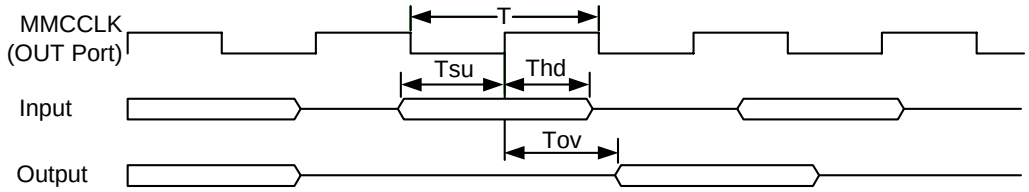


Table 2-43 lists the timing parameters of the MMC interface.

Table 2-43 Timing parameters of the MMC interface

Parameter	Symbol	Min	Typ	Max	Unit
MMCCLK clock cycle	T	20.8	–	–	ns
Input signal setup time	T_{su}	5	–	–	ns
Input signal hold time	T_{hd}	2.5	–	–	ns
Output signal delay	T_{ov}	6.5	–	14.3	ns

2.9.8 Timings of the SPI Interface

NOTE

From Figure 2-19 to Figure 2-21, the acronyms and conventions are as follows:

- MSB = most significant bit
- LSB = least significant bit
- SPICK(0):spo = 0

- SPICK(1):spo = 1

Figure 2-19 shows the SPICK timing of the SPI interface.

Figure 2-19 SPICK timing of the SPI interface

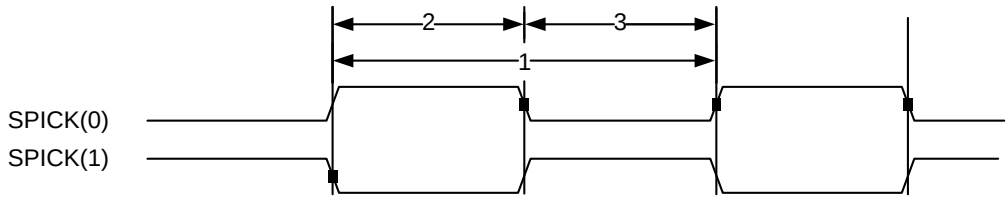


Figure 2-20 and Figure 2-21 show the timings of the SPI interface in master mode.

Figure 2-20 Timing of the SPI interface in master mode (sph = 0)

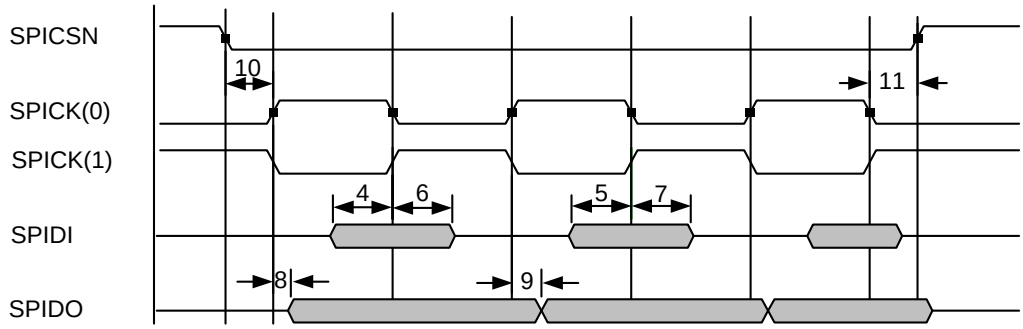


Figure 2-21 Timing of the SPI interface in master mode (sph = 1)

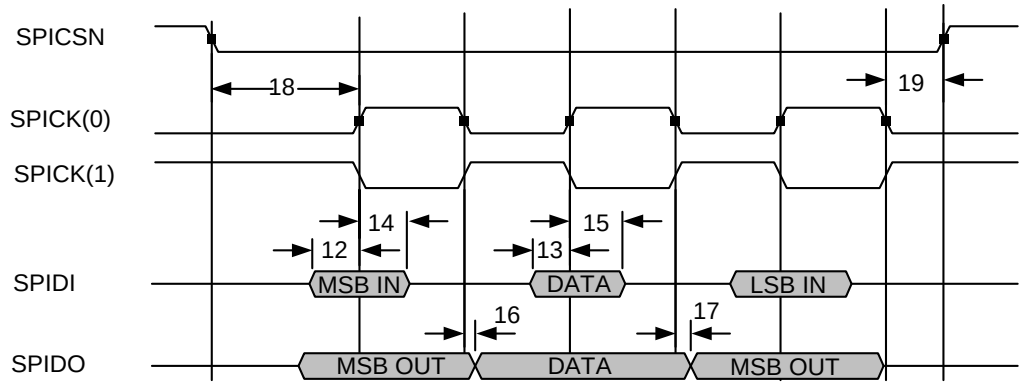


Table 2-44 lists the timing parameters of the SPI interface.

Table 2-44 Timing parameters of the SPI interface

No	Parameter	Symbol	Min	Typ	Max	Unit
1	Cycle time, SPICK	tc	–	–	–	ns

No	Parameter	Symbol	Min	Typ	Max	Unit
2	Pulse duration, SPICK high (all master modes)	tw1	–	–	–	ns
3	Pulse duration. SPICK low (all master modes)	tw2	–	–	–	ns
4	Setup time, SPIDI (input) valid before SPICK (output) falling edge	tsu1	–	–	–	ns
5	Setup time, SPIDI (in put) valid before SPICK (output) rising edge	tsu2	–	–	–	ns
6	Hold time, SPIDI (input) valid after SPICK (output) falling edge	th1	–	–	–	ns
7	Hold time, SPIDI (input) valid after SPICK (output) rising edge	th2	–	–	–	ns
8	Delay time, SPICK (output) rising edge to SPIDO (output) transition	td1	–	–	–	ns
9	Delay time, SPICK (output) falling edge to SPIDO (output) transition	td2	–	–	–	ns
10	Delay time, SPICSN (output) falling edge to first SPICK (output) rising or falling edge	td3	–	–	–	ns
11	Delay time, SPICK (output) rising or falling edge to SPICSN (output) rising edge	td4	–	–	–	ns
12	Setup time, SPIDI (input) valid before SPICK (output) rising edge	tsu3	–	–	–	ns
13	Setup time, SPIDI (in put) valid before SPICK (output) falling edge	tsu4	–	–	–	ns
14	Hold time, SPIDI (input) valid after SPICK (output) rising edge	th3	–	–	–	ns
15	Hold time, SPIDI (input) valid after SPICK (output) falling edge	th4	–	–	–	ns
16	Delay time, SSP_SPICK (output) falling edge to SSP_SPIDO (output) transition	td5	–	–	–	ns
17	Delay time, SPICK (output) rising edge to SPIDO (output) transition	td6	–	–	–	ns

No	Parameter	Symbol	Min	Typ	Max	Unit
18	Delay time, SPICSN (output) falling edge to first SPICK (output) rising or falling edge	td7	–	–	–	ns
19	Delay time, SPICK (output) rising or falling edge to SPICSN (output) rising edge	td8	–	–	–	ns

2.9.9 Timings of the UART Interface

Figure 2-22 shows the timings of the UART interface.

Figure 2-22 Timings of the UART interface

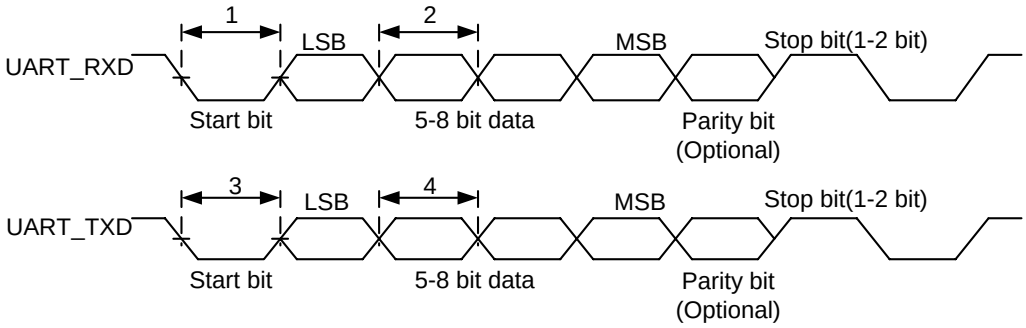


Table 2-45 lists the timing parameters of the UART receive data signal URXD.

Table 2-45 Timing parameters of the receive data signal URXD

No.	Parameter	Symbol	Min	Typ	Max	Unit
1	Pulse width, receive start bit	UART_RXD	0.96 U	–	1.05 U	ns
2	Pulse width, receive data bit	UART_RXD	0.96 U	–	1.05 U	ns

NOTE
U = UART baud time = 1/baud rate

Table 2-46 lists the timing parameters of the transmit data signal UTXD.

Table 2-46 Timing parameters of the transmit data signal UTXD

No.	Parameter	Symbol	Min	Typ	Max	Unit
3	Pulse width, transmit start bit	UART_TXD	U - 2	–	U + 2	ns

No.	Parameter	Symbol	Min	Typ	Max	Unit
4	Pulse width, transmit data bit	UART_TXD	U - 2	–	U + 2	ns

NOTE
U = UART baud time = 1/baud rate

2.9.10 Timings of the SIO Interface

Timings of the I²S Interface

Figure 2-23 shows the receive timing of the I²S interface.

Figure 2-23 Receive timing of the I²S interface

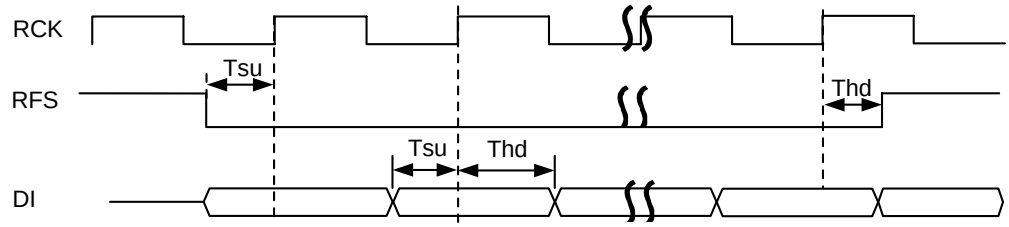


Figure 2-24 shows the transmit timing of the I²S interface.

Figure 2-24 Transmit timing of the I²S interface

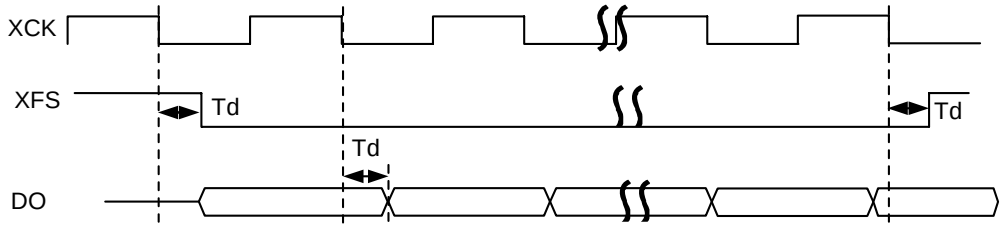


Table 2-47 lists the timing parameters of the I²S interface.

Table 2-47 Timing parameters of the I²S 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 time	T_d	0	–	8	ns

Timings of the PCM Interface

Figure 2-25 shows the receive timing of the PCM interface.

Figure 2-25 Receive timing of the PCM interface

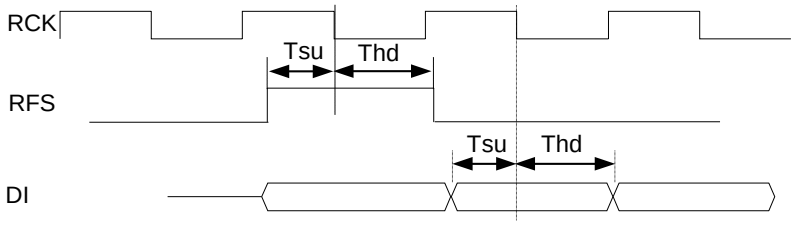


Figure 2-26 shows the transmit timing of the PCM interface.

Figure 2-26 Transmit timing of the PCM interface

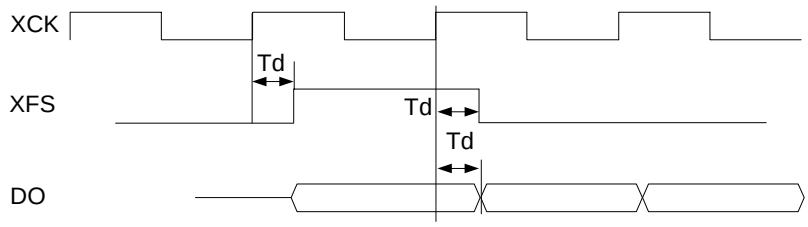


Table 2-48 lists the timing parameters of the PCM interface.

Table 2-48 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 time	T_d	0	–	8	ns

2.9.11 Timings of the SMI Interface

For details about the timings of the SMI interface, see the "Function Principle" in section 4.2.4 "Function Description."

2.9.12 Timings of the SATA Interface

The SATA interface provides high-speed external serial differential signals and has only one pair of differential data lines. The clock is multiplexed with the serial data bus. That is, there is no separate clock pins. Therefore, the peer end needs to extract the clock information when receiving data.

2.10 Package and Pin-out

2.10.1 Package

The Hi3515 uses the 0.09 µm technology and the thin fine–pitch ball grid array 441 (TFBGA441) package. The package dimensions are 19 mm x 19 mm and the ball pitch is 0.8 mm. For details about the package dimensions, see Figure 2-27 to Figure 2-31. For the dimension parameters, see Table 2-49.

Figure 2-27 Package dimensions (top view)

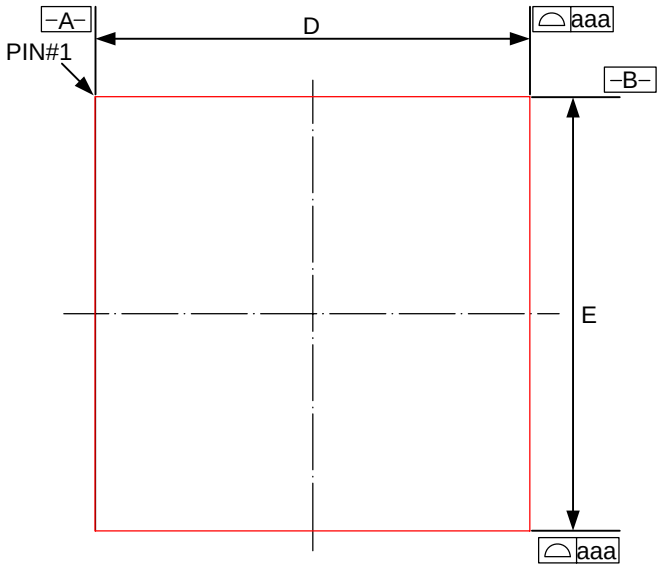


Figure 2-28 Package dimensions (bottom view)

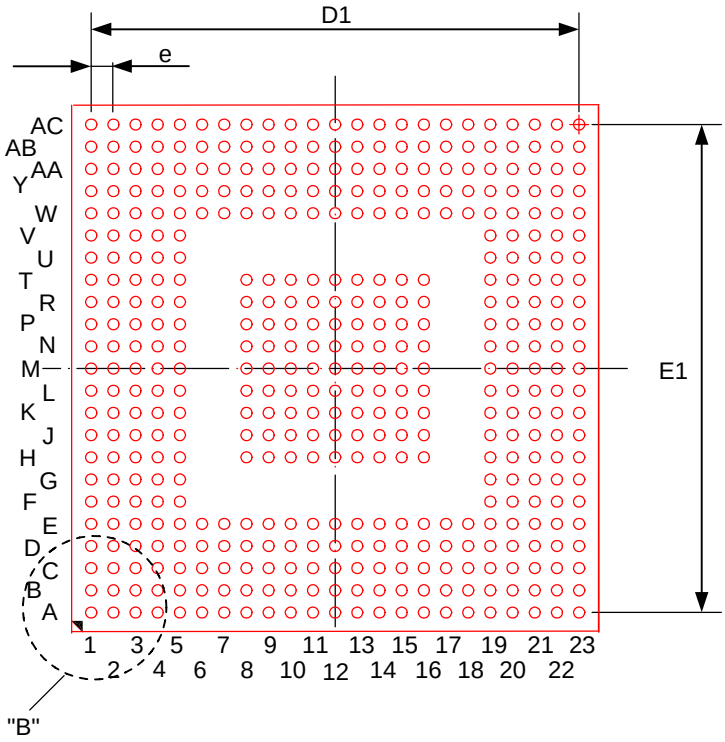


Figure 2-29 Detail B

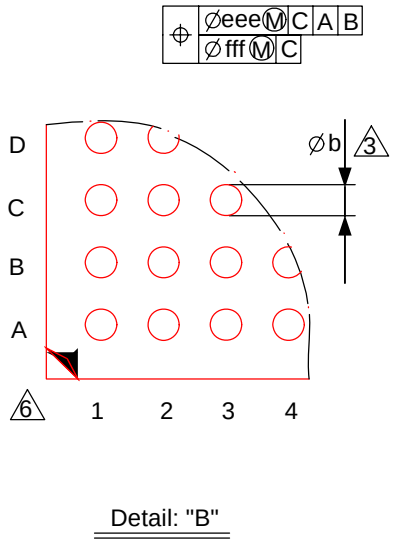


Figure 2-30 Package dimensions (side view)

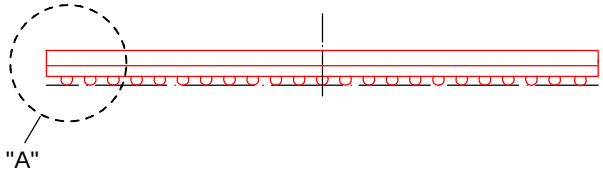


Figure 2-31 Detail A

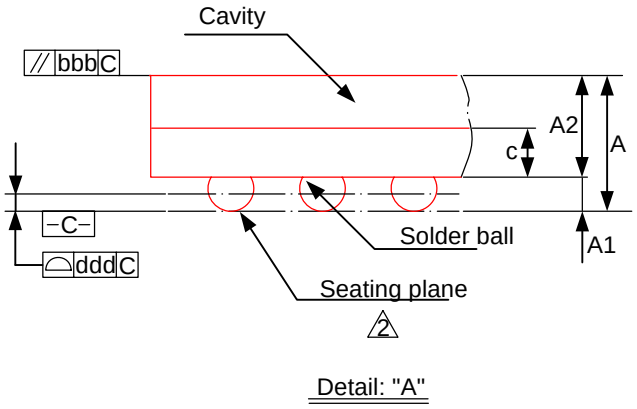


Table 2-49 Package dimensions

Parameter	Dimension (mm)		
	Min	Typ	Max
A	–	–	1.40
A1	0.25	0.30	0.35
A2	0.91	0.96	1.01
b	0.35	0.40	0.45
c	0.22	0.26	0.3
D	18.90	19.00	19.10
E	18.90	19.00	19.10
D1	–	17.60	–
E1	–	17.60	–
e	–	0.80	–
aaa	0.15		
bbb	0.20		
ddd	0.20		

Parameter	Dimension (mm)		
	Min	Typ	Max
eee	0.15		
fff	0.08		
MD/ME	23/23		

2.10.2 Pin-out

Figure 2-32 shows the pin-out (top view) of the Hi3515. Figure 2-35 to Figure 2-38 show the block diagrams.

NOTE

Figure 2-33 shows the pin-out blocks in Figure 2-35 to Figure 2-38, and Figure 2-34 shows the color definitions of the pin-out.

Figure 2-32 Pin-out

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
A	VSS	DDRDM3	DDRDQ25	DDRDQ29	DDRDQSP3	VSS	DDRDQ21	DDRDM2	DDRDQSP2	DDRADR13	DDRADR1	DDRADR3	DDRCKP1	VSS	DDRCKP0	DDRADR0	DDRCASN	DDRCSN	DDRWEN	VSS	DDRDQSN0	DDRDQ7	VSS	A
B	SIO0XFS	SIO0DI	VSS	DDRDQ30	DDRDQSN3	DDRDQ28	VSS	DDRDQ16	DDRDQSN2	DDRDQ20	DDRADR7	DDRADR5	DDRCKN1	VSS	DDRCKN0	DDRADR2	DDRRASN	DDRODT	VSS	DDRDQ5	DDRDQSP0	DDRDQ6	DDRDQ15	B
C	ACKOUT	SIO0XCK	SIO0RCK	VSS	DDRDQ24	DDRDQ26	DDRDQ23	VSS	DDRDQ22	DDRDQ17	DDRADR9	DDRADR10	DDRADR11	DDRCKE	DDRADR6	DDRBA1	DDRBA2	VSS	DDRDQ2	DDRDQ1	DDRDQ3	VSS	DDRDQ11	C
D	SIO1RFS	SIO1DI	SIO0RFS	SIO0DO	VSS	DDRDQ31	DDRDQ27	DDRDQ18	VSS	DDRDQ19	DDRADR8	VSS	DDRADR4	DDRADR12	VSS	DDRBA0	VSS	DDRDQ0	DDRDM0	DDRDQ4	VSS	DDRDQ10	DDRDQ14	D
E	VODAT2	VODAT1	VODAT0	SIO1RCK	VSS	DVDD18	DVDD18	VREF	VSS	DVDD18	DVDD18	DVDD18	VSS	VSS	DVDD18	DVDD18	VSS	VREF	VSS	VSS	DDRDQ13	DDRDQSP1	DDRDQSN1	E
F	VODAT5	VODAT4	VOCK	VODAT3	DVDD33														DVDD18	DDRDQ8	DDRDQ12	DDRDQ9	VSS	F
G	VI0HS	VODAT7	VODAT6	VSS	DVDD33														DVDD18	DDRDM1	VSS	UTXD1	UCTSN1	G
H	VI0VS	VI0CK	VI0DAT1	VI0DAT0	VSS				DVDD18	DVDD18	DVDD10	DVDD10	VSS	DVDD18	DVDD10	DVDD10	DVDD18		VSS	VSS	URXD1	URTSN1	URXD0	H
J	VI0DAT4	VI0DAT3	VI0DAT2	VSS	DVDD33				DVDD10	VSS	VSS	VSS	VSS	VSS	VSS	VSS	DVDD18		DVDD33	SDA	SCL	UTXD0	VSS	J
K	VI1DAT0	VI0DAT7	VI0DAT6	VI0DAT5	VSS				DVDD10	VSS	VSS	VSS	VSS	VSS	VSS	VSS	DVDD10		VDD10_SPLL	VSS_SPLL	AVDD33_SPLL	VSS	XOUT24	K
L	VI1DAT3	VI1DAT2	VI1DAT1	DVDD33	DVDD33				DVDD10	VSS	VSS	VSS	VSS	VSS	VSS	VSS	DVDD10		VDD10_APLL	VSS_APLL	AVDD33_APLL	TESTMODE	XIN24	L
M	VI1DAT6	VI1DAT5	VI1DAT4	VI1CK	VSS				DVDD10	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VDD10_VPLL		AVDD33_VPLL	VSS_VPLL	RSTN	WDGRSTN	VSS	M
N	VI1DAT7	VI2VS	VI2HS	VSS	DVDD33				DVDD10	VSS	VSS	VSS	VSS	VSS	VSS	VSS	SPHY_VP		VSS	TCK	TMS	TDO	TDI	N
P	VI2DAT3	VI2DAT2	VI2DAT1	VI2DAT0	VSS				AVSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	SPHY_VP		SPHY_VPH	SRESREF	TRSTN	VSS	VSS	P
R	VI2DAT5	VI2DAT4	VI2CK	DVDD33	DVDD33				AVSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	SPHY_VP		SPHY_VPH	VSS	VSS	SREFCKM	SREFCKP	R
T	VI2DAT7	VI2DAT6	VI3DAT1	VI3DAT0	AVSS				DVSS10_DAC	DVDD10_DAC	DVDD10	USBVDD	USBVSS	DVDD10	DVDD10	DVDD10	DVDD10		SPHY_VPH	STXP1	STXM1	VSS	VSS	T
U	VI3DAT2	VI3DAT3	VI3CK	VI3DAT4	AVSS														VSS	VSS	VSS	SRXM1	SRXP1	U
V	VI3DAT5	VI3DAT6	VI3DAT7	AVDD33	AVDD33														STXM0	STXP0	VSS	SRXM0	SRXP0	V
W	VREFINDAC0	COMPDAC0	AVDD33	AVDD33	DVSS33_DAC	DVDD33_DAC	VSS	DVDD33	USBVSSA33	USBREXT	USBVDDA33	VSS	DVDD33	DVDD33	VSS	DVDD33	DVDD33	VSS	SPHY_VPH	SPHY_VPH	VSS	VSS	VSS	W
Y	AVSS	RSETDAC0	RSETDAC1	VREFINDAC1	MDCK	ETXD2	ERXCK	ERXD3	USBVSSA33	USBDM1	USBVDDA33	DVDD33	EBIDQ6	EBIADR2	VSS	EBIADR9	SMIOEN	VSS	SMICS1N	NFCLE	NFOEN	NFCS0N	NFCS1N	Y
AA	DACVGA0R	AVSS	COMPDAC1	VGAHS	MDIO	ETXCK	ECRS	ERXD2	USBVSSA33	USBDP1	USBVDDA33	EBIDQ2	EBIDQ5	EBIADR1	EBIADR5	EBIADR8	EBIADR12	EBIADR24	SMICS0N	EBIWEN	EBIRDYN	NFALE	NFRB	AA
AB	DACVGA1R	AVSS	AVSS	VGAVS	ETXD0	ETXD3	ECOL	ERXD1	VSS	USBDM0	VSS	EBIDQ1	EBIDQ4	EBIADR0	EBIADR4	EBIADR7	EBIADR11	EBIADR14	EBIADR16	EBIADR18	EBIADR20	EBIADR22	EBIADR23	AB
AC	AVSS	DACVGA1G	DACVGA1B	AVSS	ETXD1	ERXDV	ETXEN	ERXD0	VSS	USBDP0	VSS	EBIDQ0	EBIDQ3	EBIDQ7	EBIADR3	EBIADR6	EBIADR10	EBIADR13	EBIADR15	EBIADR17	EBIADR19	EBIADR21	VSS	AC
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	

Figure 2-33 Pin-out blocks

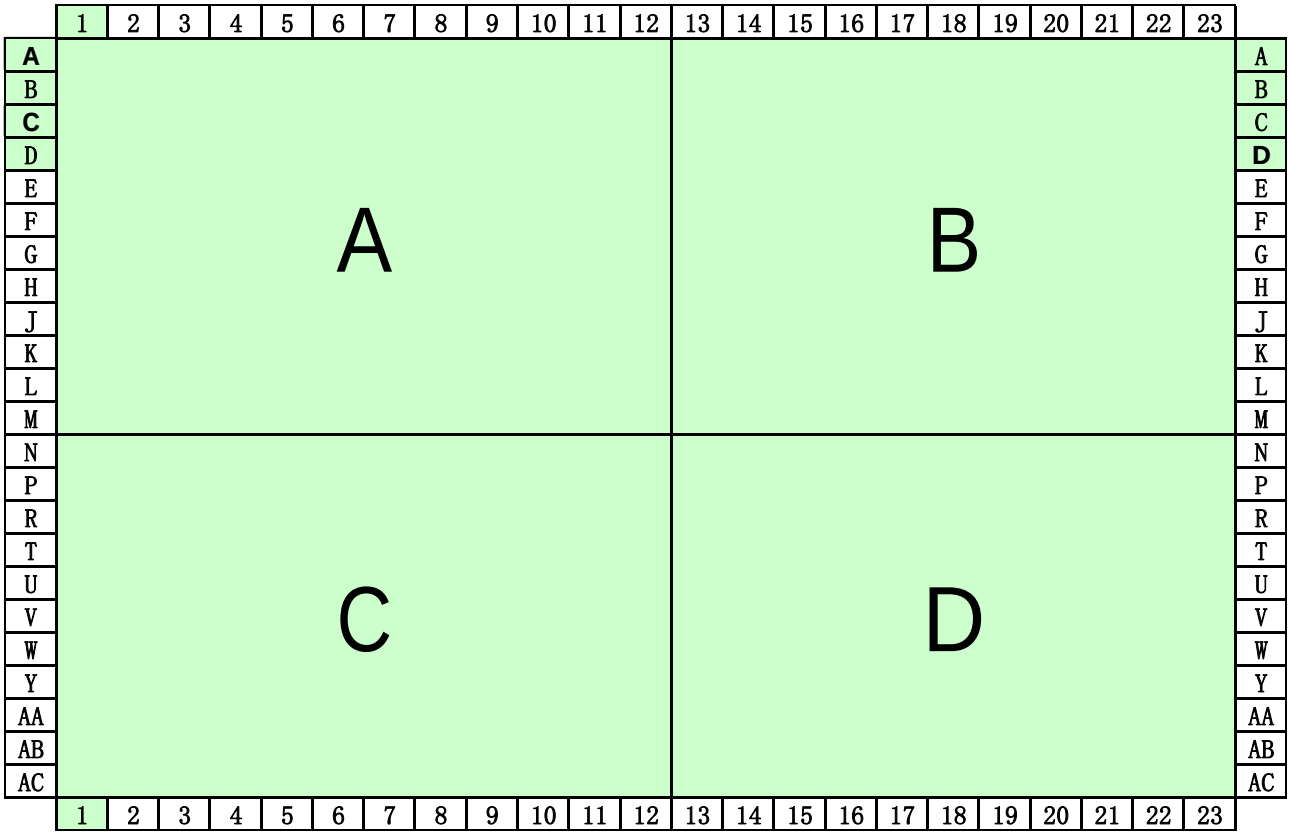


Figure 2-34 Color definitions of the pin-out

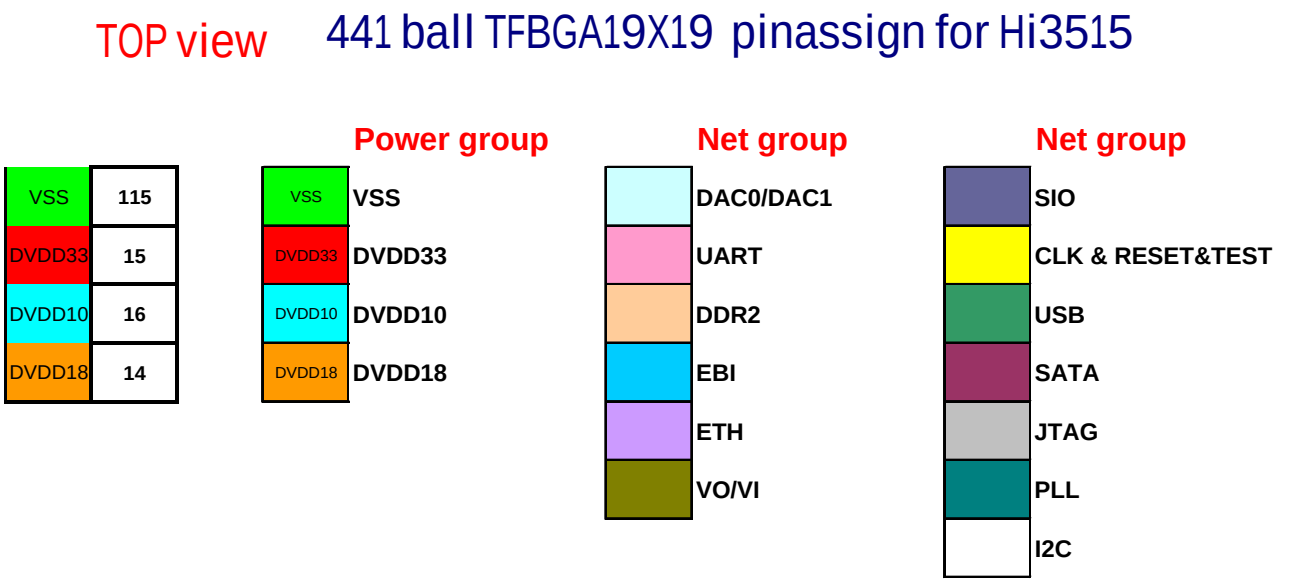


Figure 2-35 Hi3515 pin-out block A

	1	2	3	4	5	6	7	8	9	10	11	12
A	VSS	DDRDM3	DDRDQ25	DDRDQ29	DDRDQSP3	VSS	DDRDQ21	DDRDM2	DDRDQSP2	DDRADR13	DDRADR1	DDRADR3
B	SIO0XFS	SIO0DI	VSS	DDRDQ30	DDRDQSN3	DDRDQ28	VSS	DDRDQ16	DDRDQSN2	DDRDQ20	DDRADR7	DDRADR5
C	ACKOUT	SIO0XCK	SIO0RCK	VSS	DDRDQ24	DDRDQ26	DDRDQ23	VSS	DDRDQ22	DDRDQ17	DDRADR9	DDRADR10
D	SIO1RFS	SIO1DI	SIO0RFS	SIO0DO	VSS	DDRDQ31	DDRDQ27	DDRDQ18	VSS	DDRDQ19	DDRADR8	VSS
E	VODAT2	VODAT1	VODAT0	SIO1RCK	VSS	DVDD18	DVDD18	VREF	VSS	DVDD18	DVDD18	DVDD18
F	VODAT5	VODAT4	VOCK	VODAT3	DVDD33							
G	VI0HS	VODAT7	VODAT6	VSS	DVDD33							
H	VI0VS	VI0CK	VI0DAT1	VI0DAT0	VSS			DVDD18	DVDD18	DVDD10	DVDD10	VSS
J	VI0DAT4	VI0DAT3	VI0DAT2	VSS	DVDD33			DVDD10	VSS	VSS	VSS	VSS
K	VI1DAT0	VI0DAT7	VI0DAT6	VI0DAT5	VSS			DVDD10	VSS	VSS	VSS	VSS
L	VI1DAT3	VI1DAT2	VI1DAT1	DVDD33	DVDD33			DVDD10	VSS	VSS	VSS	VSS
M	VI1DAT6	VI1DAT5	VI1DAT4	VI1CK	VSS			DVDD10	VSS	VSS	VSS	VSS

Figure 2-36 Hi3515 pin-out block B

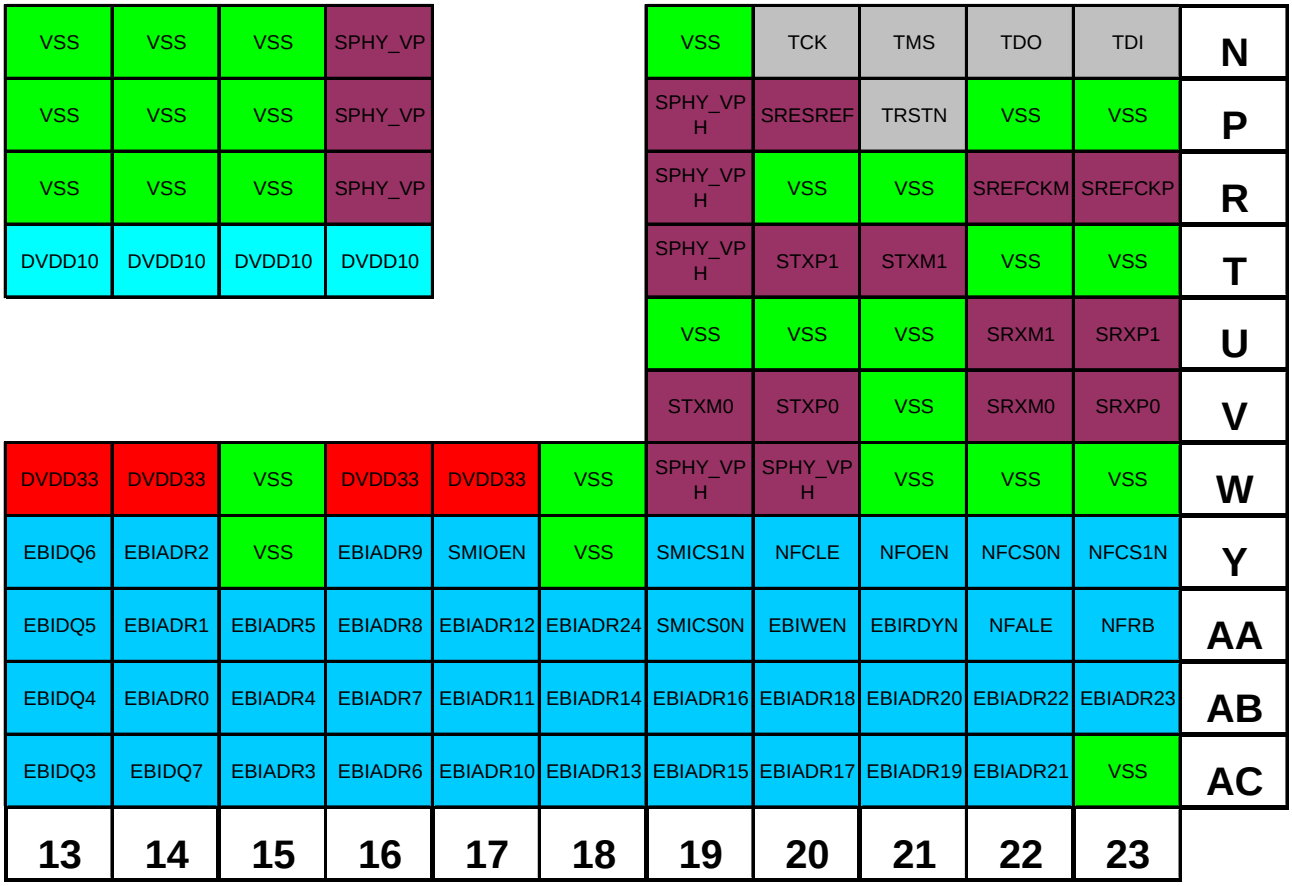
13	14	15	16	17	18	19	20	21	22	23	
DDRCKP1	VSS	DDRCKP0	DDRADR0	DDRCASN	DDRCSN	DDRWEN	VSS	DDRDQSN0	DDRDQ7	VSS	A
DDRCKN1	VSS	DDRCKN0	DDRADR2	DDRRASN	DDRODT	VSS	DDRDQ5	DDRDQSP0	DDRDQ6	DDRDQ15	B
DDRADR11	DDRCKE	DDRADR6	DDRBA1	DDRBA2	VSS	DDRDQ2	DDRDQ1	DDRDQ3	VSS	DDRDQ11	C
DDRADR4	DDRADR12	VSS	DDRBA0	VSS	DDRDQ0	DDRDM0	DDRDQ4	VSS	DDRDQ10	DDRDQ14	D
VSS	VSS	DVDD18	DVDD18	VSS	VREF	VSS	VSS	DDRDQ13	DDRDQSP1	DDRDQSN1	E
						DVDD18	DDRDQ8	DDRDQ12	DDRDQ9	VSS	F
						DVDD18	DDRDM1	VSS	UTXD1	UCTSN1	G
DVDD18	DVDD10	DVDD10	DVDD18			VSS	VSS	URXD1	URTSN1	URXD0	H
VSS	VSS	VSS	DVDD18			DVDD33	SDA	SCL	UTXD0	VSS	J
VSS	VSS	VSS	DVDD10			VDD10_SPLL	VSS_SPLL	AVDD33_SPLL	VSS	XOUT24	K
VSS	VSS	VSS	DVDD10			VDD10_APLL	VSS_APLL	AVDD33_APLL	TESTMODE	XIN24	L
VSS	VSS	VSS	VDD10_VPLL			AVDD33_VPLL	VSS_VPLL	RSTN	WDGRSTN	VSS	M

Figure 2-37 Hi3515 pin-out block C

	1	2	3	4	5	6	7	8	9	10	11	12
N	VI1DAT7	VI2VS	VI2HS	VSS	DVDD33							
P	VI2DAT3	VI2DAT2	VI2DAT1	VI2DAT0	VSS							
R	VI2DAT5	VI2DAT4	VI2CK	DVDD33	DVDD33							
T	VI2DAT7	VI2DAT6	VI3DAT1	VI3DAT0	AVSS							
U	VI3DAT2	VI3DAT3	VI3CK	VI3DAT4	AVSS							
V	VI3DAT5	VI3DAT6	VI3DAT7	AVDD33	AVDD33							
W	VREFINDAC0	COMPDAC0	AVDD33	AVDD33	DVSS33_DAC	DVDD33_DAC	VSS	DVDD33	USBVSSA33	USBREXT	USBVDDA33	VSS
Y	AVSS	RSETDAC0	RSETDAC1	VREFINDAC1	MDCK	ETXD2	ERXCK	ERXD3	USBVSSA33	USBDM1	USBVDDA33	DVDD33
AA	DACVGA0R	AVSS	COMPDAC1	VGAHS	MDIO	ETXCK	ECRS	ERXD2	USBVSSA33	USBDP1	USBVDDA33	EBIDQ2
AB	DACVGA1R	AVSS	AVSS	VGAVS	ETXD0	ETXD3	ECOL	ERXD1	VSS	USBDM0	VSS	EBIDQ1
AC	AVSS	DACVGA1G	DACVGA1B	AVSS	ETXD1	ERXDV	ETXEN	ERXD0	VSS	USBDP0	VSS	EBIDQ0

DVDD10	VSS	VSS	VSS	VSS
AVSS	VSS	VSS	VSS	VSS
AVSS	VSS	VSS	VSS	VSS
DVSS10_DAC	DVDD10_DAC	DVDD10	USBVDD	USBVSS

Figure 2-38 Hi3515 pin-out block D



2.10.3 Pin Arrangement Table

Table 2-50 lists the pins of the Hi3515 in order.

Table 2-50 Pin arrangement table

Position	Pin	Position	Pin
A1	VSS	M13	VSS
A2	DDRDM3	M14	VSS
A3	DDRDQ25	M15	VSS
A4	DDRDQ29	M16	VDD10_VPLL
A5	DDRDQSP3	M19	AVDD33_VPLL
A6	VSS	M20	VSS_VPLL
A7	DDRDQ21	M21	RSTN
A8	DDRDM2	M22	WDGRSTN
A9	DDRDQSP2	M23	VSS

Position	Pin	Position	Pin
A10	DDRADR13	N1	VI1DAT7
A11	DDRADR1	N2	VI2VS
A12	DDRADR3	N3	VI2HS
A13	DDRCKP1	N4	VSS
A14	VSS	N5	DVDD33
A15	DDRCKP0	N8	DVDD10
A16	DDRADR0	N9	VSS
A17	DDRCASN	N10	VSS
A18	DDRCSN	N11	VSS
A19	DDRWEN	N12	VSS
A20	VSS	N13	VSS
A21	DDRDQSN0	N14	VSS
A22	DDRDQ7	N15	VSS
A23	VSS	N16	SPHY_VP
B1	SIO0XFS	N19	VSS
B2	SIO0DI	N20	TCK
B3	VSS	N21	TMS
B4	DDRDQ30	N22	TDO
B5	DDRDQSN3	N23	TDI
B6	DDRDQ28	P1	VI2DAT3
B7	VSS	P2	VI2DAT2
B8	DDRDQ16	P3	VI2DAT1
B9	DDRDQSN2	P4	VI2DAT0
B10	DDRDQ20	P5	VSS
B11	DDRADR7	P8	AVSS
B12	DDRADR5	P9	VSS
B13	DDRCKN1	P10	VSS
B14	VSS	P11	VSS
B15	DDRCKN0	P12	VSS
B16	DDRADR2	P13	VSS
B17	DDRRASN	P14	VSS

Position	Pin	Position	Pin
B18	DDRODT	P15	VSS
B19	VSS	P16	SPHY_VP
B20	DDRDQ5	P19	SPHY_VPH
B21	DDRDQSP0	P20	SRESREF
B22	DDRDQ6	P21	TRSTN
B23	DDRDQ15	P22	VSS
C1	ACKOUT	P23	VSS
C2	SIO0XCK	R1	VI2DAT5
C3	SIO0RCK	R2	VI2DAT4
C4	VSS	R3	VI2CK
C5	DDRDQ24	R4	DVDD33
C6	DDRDQ26	R5	DVDD33
C7	DDRDQ23	R8	AVSS
C8	VSS	R9	VSS
C9	DDRDQ22	R10	VSS
C10	DDRDQ17	R11	VSS
C11	DDRADR9	R12	VSS
C12	DDRADR10	R13	VSS
C13	DDRADR11	R14	VSS
C14	DDRCKE	R15	VSS
C15	DDRADR6	R16	SPHY_VP
C16	DDRBA1	R19	SPHY_VPH
C17	DDRBA2	R20	VSS
C18	VSS	R21	VSS
C19	DDRDQ2	R22	SREFCKM
C20	DDRDQ1	R23	SREFCKP
C21	DDRDQ3	T1	VI2DAT7
C22	VSS	T2	VI2DAT6
C23	DDRDQ11	T3	VI3DAT1
D1	SIO1RFS	T4	VI3DAT0
D2	SIO1DI	T5	AVSS

Position	Pin	Position	Pin
D3	SIO0RFS	T8	DVSS10_DAC
D4	SIO0DO	T9	DVDD10_DAC
D5	VSS	T10	DVDD10
D6	DDRDQ31	T11	USBVDD
D7	DDRDQ27	T12	USBVSS
D8	DDRDQ18	T13	DVDD10
D9	VSS	T14	DVDD10
D10	DDRDQ19	T15	DVDD10
D11	DDRADR8	T16	DVDD10
D12	VSS	T19	SPHY_VPH
D13	DDRADR4	T20	STXP1
D14	DDRADR12	T21	STXM1
D15	VSS	T22	VSS
D16	DDRBA0	T23	VSS
D17	VSS	U1	VI3DAT2
D18	DDRDQ0	U2	VI3DAT3
D19	DDRDM0	U3	VI3CK
D20	DDRDQ4	U4	VI3DAT4
D21	VSS	U5	AVSS
D22	DDRDQ10	U19	VSS
D23	DDRDQ14	U20	VSS
E1	VODAT2	U21	VSS
E2	VODAT1	U22	SRXM1
E3	VODAT0	U23	SRXP1
E4	SIO1RCK	V1	VI3DAT5
E5	VSS	V2	VI3DAT6
E6	DVDD18	V3	VI3DAT7
E7	DVDD18	V4	AVDD33
E8	VREF	V5	AVDD33
E9	VSS	V19	STXM0
E10	DVDD18	V20	STXP0

Position	Pin	Position	Pin
E11	DVDD18	V21	VSS
E12	DVDD18	V22	SRXM0
E13	VSS	V23	SRXP0
E14	VSS	W1	VREFINDAC0
E15	DVDD18	W2	COMPDAC0
E16	DVDD18	W3	AVDD33
E17	VSS	W4	AVDD33
E18	VREF	W5	DVSS33_DAC
E19	VSS	W6	DVDD33_DAC
E20	VSS	W7	VSS
E21	DDRDQ13	W8	DVDD33
E22	DDRDQSP1	W9	USBVSSA33
E23	DDRDQSN1	W10	USBREXT
F1	VODAT5	W11	USBVDDA33
F2	VODAT4	W12	VSS
F3	VOCK	W13	DVDD33
F4	VODAT3	W14	DVDD33
F5	DVDD33	W15	VSS
F19	DVDD18	W16	DVDD33
F20	DDRDQ8	W17	DVDD33
F21	DDRDQ12	W18	VSS
F22	DDRDQ9	W19	SPHY_VPH
F23	VSS	W20	SPHY_VPH
G1	VI0HS	W21	VSS
G2	VODAT7	W22	VSS
G3	VODAT6	W23	VSS
G4	VSS	Y1	AVSS
G5	DVDD33	Y2	RSETDAC0
G19	DVDD18	Y3	RSETDAC1
G20	DDRDM1	Y4	VREFINDAC1
G21	VSS	Y5	MDCK

Position	Pin	Position	Pin
G22	UTXD1	Y6	ETXD2
G23	UCTSN1	Y7	ERXCK
H1	VI0VS	Y8	ERXD3
H2	VI0CK	Y9	USBVSSA33
H3	VI0DAT1	Y10	USBDM1
H4	VI0DAT0	Y11	USBVDDA33
H5	VSS	Y12	DVDD33
H8	DVDD18	Y13	EBIDQ6
H9	DVDD18	Y14	EBIADR2
H10	DVDD10	Y15	VSS
H11	DVDD10	Y16	EBIADR9
H12	VSS	Y17	SMIOEN
H13	DVDD18	Y18	VSS
H14	DVDD10	Y19	SMICS1N
H15	DVDD10	Y20	NFCLE
H16	DVDD18	Y21	NFOEN
H19	VSS	Y22	NFCS0N
H20	VSS	Y23	NFCS1N
H21	URXD1	AA1	DACVGA0R
H22	URTSN1	AA2	AVSS
H23	URXD0	AA3	COMPDAC1
J1	VI0DAT4	AA4	VGAHS
J2	VI0DAT3	AA5	MDIO
J3	VI0DAT2	AA6	ETXCK
J4	VSS	AA7	ECRS
J5	DVDD33	AA8	ERXD2
J8	DVDD10	AA9	USBVSSA33
J9	VSS	AA10	USBDP1
J10	VSS	AA11	USBVDDA33
J11	VSS	AA12	EBIDQ2
J12	VSS	AA13	EBIDQ5

Position	Pin	Position	Pin
J13	VSS	AA14	EBIADR1
J14	VSS	AA15	EBIADR5
J15	VSS	AA16	EBIADR8
J16	DVDD18	AA17	EBIADR12
J19	DVDD33	AA18	EBIADR24
J20	SDA	AA19	SMICS0N
J21	SCL	AA20	EBIWEN
J22	UTXD0	AA21	EBIRDYN
J23	VSS	AA22	NFALE
K1	VI1DAT0	AA23	NFRB
K2	VI0DAT7	AB1	DACVGA1R
K3	VI0DAT6	AB2	AVSS
K4	VI0DAT5	AB3	AVSS
K5	VSS	AB4	VGAVS
K8	DVDD10	AB5	ETXD0
K9	VSS	AB6	ETXD3
K10	VSS	AB7	ECOL
K11	VSS	AB8	ERXD1
K12	VSS	AB9	VSS
K13	VSS	AB10	USBDM0
K14	VSS	AB11	VSS
K15	VSS	AB12	EBIDQ1
K16	DVDD10	AB13	EBIDQ4
K19	VDD10_SPLL	AB14	EBIADR0
K20	VSS_SPLL	AB15	EBIADR4
K21	AVDD33_SPLL	AB16	EBIADR7
K22	VSS	AB17	EBIADR11
K23	XOUT24	AB18	EBIADR14
L1	VI1DAT3	AB19	EBIADR16
L2	VI1DAT2	AB20	EBIADR18
L3	VI1DAT1	AB21	EBIADR20

Position	Pin	Position	Pin
L4	DVDD33	AB22	EBIADR22
L5	DVDD33	AB23	EBIADR23
L8	DVDD10	AC1	AVSS
L9	VSS	AC2	DACVGA1G
L10	VSS	AC3	DACVGA1B
L11	VSS	AC4	AVSS
L12	VSS	AC5	ETXD1
L13	VSS	AC6	ERXDV
L14	VSS	AC7	ETXEN
L15	VSS	AC8	ERXD0
L16	DVDD10	AC9	VSS
L19	VDD10_APLL	AC10	USBDP0
L20	VSS_APLL	AC11	VSS
L21	AVDD33_APLL	AC12	EBIDQ0
L22	TESTMODE	AC13	EBIDQ3
L23	XIN24	AC14	EBIDQ7
M1	VI1DAT6	AC15	EBIADR3
M2	VI1DAT5	AC16	EBIADR6
M3	VI1DAT4	AC17	EBIADR10
M4	VI1CK	AC18	EBIADR13
M5	VSS	AC19	EBIADR15
M8	DVDD10	AC20	EBIADR17
M9	VSS	AC21	EBIADR19
M10	VSS	AC22	EBIADR21
M11	VSS	AC23	VSS
M12	VSS	–	–

Contents

Figures

Tables

3 System

3.1 Reset

3.1.1 Overview

The reset management module manages the reset of the entire Hi3515 and all functional modules as follows:

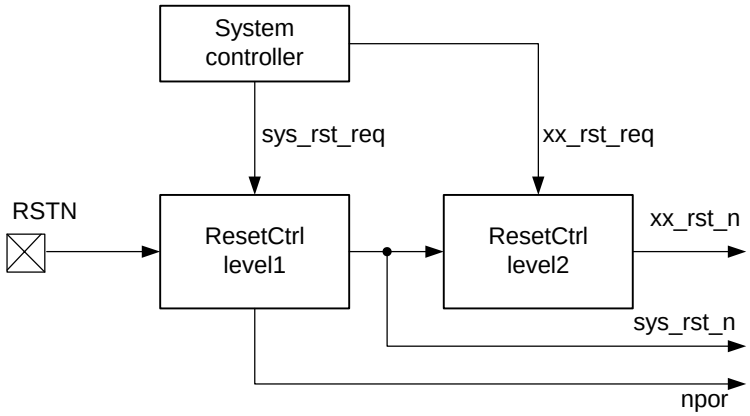
- Manages and controls power-on reset.
- Controls the system soft reset and the separate soft reset of each functional module.
- Synchronizes reset signals to the clock domain corresponding to each module.

The reset management module generates reset signals for each internal functional module.

3.1.2 Reset Control

Figure 3-1 shows the control diagram of reset signals.

Figure 3-1 Control diagram of reset signals



RSTN: Power-on reset signal derived from the RSTN pin of the Hi3515.
sys_rst_req: Global soft reset request signal derived from the system controller.
xx_rst_req: Separate soft reset request signal of each submodule derived from the system controller.
xx_rst_n, sys_rst_n, and npor: reset signals.

Table 3-1 Types of reset signals

Type	Source	Function
Global hard reset signal (npor)	Derived from the RSTN reset pin.	Globally resets the entire Hi3515.
Global soft reset signal (syn_rst_n)	Derived from the global soft reset register of the software configuration system controller.	Globally resets all the modules of the Hi3515 excluding the clock reset circuit and the test circuit.
Submodule reset signal (xx_rst_n)	Derived from the submodule reset control register of the software configuration system controller.	Separately resets each submodule of the Hi3515.

3.1.3 Reset Configuration

Power-on Reset

The RSTN is the functional reset input/output (IO) of the Hi3515. To implement the power-on reset, the following conditions must be met:

- The power-on reset IO pin inputs a low-level pulse.
- The clock input by the XIN pin of the crystal oscillator clock works properly.
- The input power-on reset signals are maintained as low for more than 12 XIN crystal oscillator clock cycles.

System Reset

The system is reset in either of the following two ways:

- Power-on reset
- Global soft reset, controlled by the system controller SC_SYSSTAT.

Soft Reset

Soft reset is controlled by configuring the associated system controller. For details about the configurations, see the descriptions of SC_SYSSTAT.

CAUTION

- After a system soft reset request is sent, the circuit reset is cleared after at least 360 system clock cycles.
- The separate soft reset of each module is not automatically cleared. For example, if a module is reset after 1 is written to the associated bit, the reset of this module is cleared only when the associated bit is set to 0.
- The USB 2.0 host physical layer entity sublayer (PHY) module, MMC/SD/SDIO controller (MMC), and Ethernet MAC (ETH), video input (VI), video output (VO), and real time clock (RTC) modules are reset by default when they are powered on. Therefore, the resets must be cleared before the modules work.

3.2 Clock

3.2.1 Overview

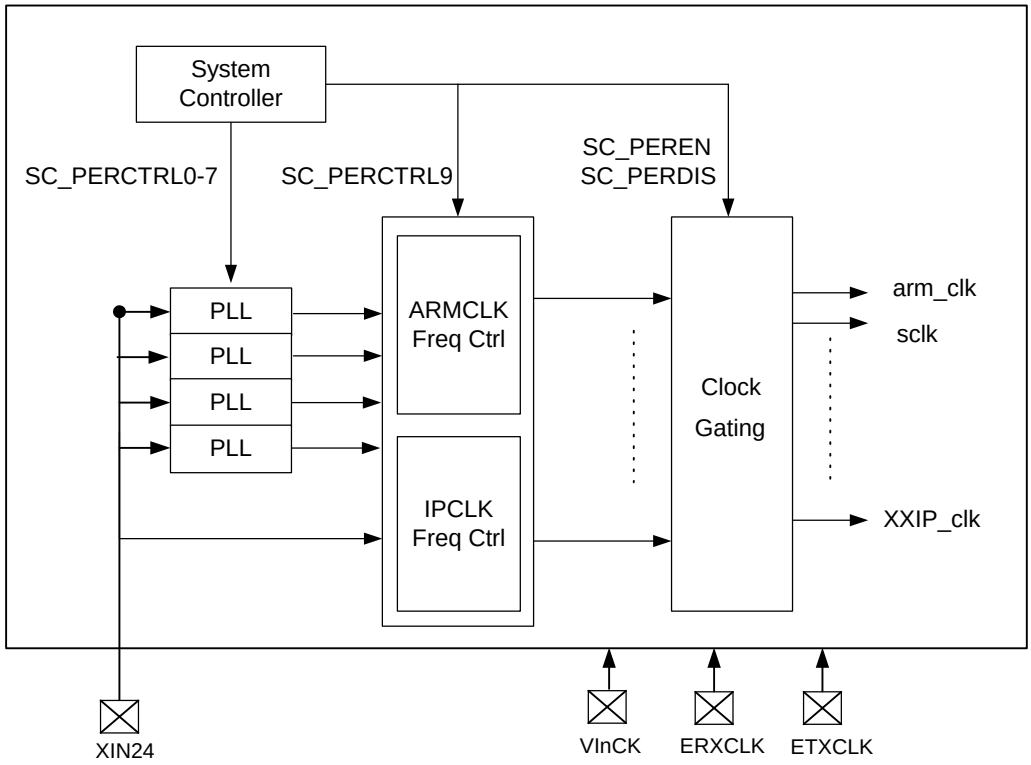
The clock management module manages clock input, clock generation, and clock control in a unified manner as follows:

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

3.2.2 Clock Control

Figure 3-2 shows the functional block diagram of the clock management module.

Figure 3-2 Functional block diagram of the clock management module



Note: In this diagram, the value of n in VInCK ranges from 0 to 3.

The inputs of the clock management module are as follows:

- Clock inputs from the pins: XIN24, VI0CK, VI1CK, VI2CK, VI3CK, ERXCLK, and ETXCLK.
  - XIN24 is the input clock of the phase-locked loop (PLL) and USB 2.0 host PHY.
  - VI0CK, VI1CK, VI2CK, and VI3CK are video input clocks.
  - ERXCLK and ETXCLK are the interface clocks of the ETH module.
- Clock control registers derived from the system controller.
  - PLL frequency configuration
  - IP clock frequency configuration
  - Clock Gating Configuration

The clock management module consist of the following parts:

- PLL unit: Generates ARM clock, bus clock, and the clocks required by other peripherals.
- ARM frequency control unit (ARMCLK Freq Ctrl) and module clock frequency control unit (IPCLK Freq Ctrl)
- Clock gating management unit

3.2.3 Clock Configuration

PLL Configuration Registers

The Hi3515 has four internal PLLs. Each PLL uses two sets of configuration registers. See Table 3-2. For details about each configuration register, see the descriptions of the system controllers SC_PERCTRL0 to SC_PERCTRL7.

Table 3-2 PLL configuration registers of the Hi3515

PLL	Configuration Register	Description
APLL	SC_PERCRTL0	The APLL is used to generate ARMCLK and bus clock.
	SC_PERCRTL1	
EPLL	SC_PERCRTL2	The EPLL is used to generate the ETH clock.
	SC_PERCRTL3	
VPLL0	SC_PERCRTL4	VPLL0 is used to generate the clock of the VO0 module.
	SC_PERCRTL5	
VPLL1	SC_PERCRTL6	VPLL1 is used to generate the clock of the VO1 module.
	SC_PERCRTL7	

All the PLLs use the crystal oscillator clock input by the XIN24 pin as the input clock (24 MHz). Each of the PLLs can provide multiple clock frequencies. See Table 3-3.

Table 3-3 Formulas for calculating the frequency of each PLL pin of the Hi3515

PLL Pin	Formula	Precautions
FREF	24 MHz	The Hi3515 must be connected to a 24 MHz crystal oscillator clock.
FOUTVCO	FREF x (fbdiv + frac/2^24)/refdiv	The operating frequency of the PLL voltage controlled oscillator (VCO) must range from 800 MHz to 2.4 GHz. When the PLL frequency divider is an integer, frac is equal to 0.
FOUTPOSTDIV	FOUTVCO/pstdiv1 x pstdiv2	–
FOUT1ph0	FOUTVCO/(pstdiv1 x 2)	–
FOUT2	FOUTVCO/(pstdiv1 x 4)	–
FOUT3	FOUTVCO/(pstdiv1 x 6)	–
FOUT4	FOUTVCO/(pstdiv1 x 8)	–

The configured values of the APLL and EPLL are fixed, whereas the values of VPLL0 and VPLL1 can be configured according to application scenarios.

The APLL uses FOUT2 and FOUT3. In typical application scenario, the required frequencies of FOUT2 and FOUT3 are respectively 400 MHz and 266 MHz respectively. Table 3-4 describes the recommended configuration, that is, SC_PERCTRL0 = 0x8900_0000 and SC_PERCTRL1 = 0x006C_30C8.

Table 3-4 Configuration parameters of the Hi3515 APLL

Bit	PLL Pin	Description	Recommended Value
SC_PERCTRL0[31]	dsmpd	0: decimal frequency-division mode 1: integer frequency-division mode	0b1
SC_PERCTRL0 [30]	bypass	0: no bypass 1: bypass	0b0
SC_PERCTRL0[29:27]	postdiv2	Level-2 output frequency divider of the PLL	0b001
SC_PERCTRL0[26:24]	postdiv1	Level-1 output frequency divider of the PLL	0b001
SC_PERCTRL0[23:0]	frac	Decimal frequency divider of the PLL	0x00_0000
SC_PERCTRL1[22]	reset	PLL reset control 0: reset 1: not reset	0b1
SC_PERCTRL1[21]	pd	PLL power down control 0: disabled 1: enabled	0b1
SC_PERCTRL1[20]	foutvcopd	PLL VCO output power down control 0: disabled 1: enabled	0b0
SC_PERCTRL1[19]	postdivpd	APLL POSTDIV output power down control 0: disabled 1: enabled	0b1
SC_PERCTRL1[18]	fout4phasepd	PLL FOUT output power down control 0: disabled 1: enabled	0b1

Bit	PLL Pin	Description	Recommended Value
SC_PERCTRL1[17:12]	refdiv	Frequency divider of the PLL reference clock	0x03
SC_PERCTRL1[11:0]	fbdiv	Integer frequency multiplier of the PLL	0x0C8

The EPLL uses FOUT1ph0, FOUT2, and FOUT4. In typical application scenario, their required frequencies are respectively 500 MHz, 250 MHz, and 125 MHz. Table 3-5 describes the recommended configuration, that is, SC_PERCTRL6 = 0xA100_0000 and SC_PERCTRL7 = 0x006C_307D.

Table 3-5 Configuration parameters of the Hi3515 EPLL

Bit	PLL Pin	Description	Recommended Value
SC_PERCTRL6[31]	dsmpd	0: Decimal frequency-division mode 1: Integer frequency-division mode	0x1
SC_PERCTRL6[30]	bypass	0: no bypass 1: bypass	0x0
SC_PERCTRL6[29:27]	postdiv2	Level-2 output frequency divider of the PLL	0x4
SC_PERCTRL6[26:24]	postdiv1	Level-1 output frequency divider of the PLL	0x1
SC_PERCTRL6[23:0]	frac	Decimal frequency divider of the PLL	0x00_0000
SC_PERCTRL7[22]	reset	PLL reset control 0: reset 1: not reset	0x1
SC_PERCTRL7[21]	pd	PLL power down control 0: disabled 1: enabled	0x1
SC_PERCTRL7[20]	foutvcopd	PLL VCO output power down control 0: disabled 1: enabled	0x0

Bit	PLL Pin	Description	Recommended Value
SC_PERCTRL7[19]	postdivpd	APLL POSTDIV output power down control 0: disabled 1: enabled	0x1
SC_PERCTRL7[18]	vfout4phasepd	PLL FOUT output power down control 0: disabled 1: enabled	0x1
SC_PERCTRL7[17:12]	refdiv	Frequency divider of the PLL reference clock	0x3
SC_PERCTRL7[11:0]	fbdiv	Integer frequency multiplier of the PLL	0x07D

Both VPLL0 and VPLL1 use FOUTPOSTDIV of the PLL as the frequency output pin. The required frequencies can be configured through SC_PERCTRL2 to SC_PERCTRL5. Table 3-6 and Table 3-7 show the recommended configuration.

Table 3-6 Configuration parameters of the Hi3515 VPLL0

Bit	PLL Pin	Description	Recommended Value
SC_PERCTRL2[31]	dsmpd	0: Decimal frequency-division mode 1: Integer frequency-division mode	Depending on the application scenario
SC_PERCTRL2[30]	bypass	0: no bypass 1: bypass	0b0
SC_PERCTRL2[29:27]	postdiv2	Level-2 output frequency divider of the PLL	Depending on the application scenario
SC_PERCTRL2[26:24]	postdiv1	Level-1 output frequency divider of the PLL	Depending on the application scenario
SC_PERCTRL2 [23:0]	frac	Decimal frequency divider of the PLL	Depending on the application scenario
SC_PERCTRL3[22]	reset	PLL reset control 0: reset 1: not reset	0b1

Bit	PLL Pin	Description	Recommended Value
SC_PERCTRL3 [21]	pd	PLL power down control 0: disabled 1: enabled	0b1
SC_PERCTRL3 [20]	foutvcopd	PLL VCO output power down control 0: disabled 1: enabled	0b0
SC_PERCTRL3[19]	postdivpd	APLL POSTDIV output power down control 0: disabled 1: enabled	0b1
SC_PERCTRL3[18]	vfout4phasepd	PLL FOUT output power down control 0: disabled 1: enabled	0b1
SC_PERCTRL3[17:12]	refdiv	Frequency divider of the PLL reference clock	Depending on the application scenario
SC_PERCTRL3 [11:0]	fbdiv	Integer frequency multiplier of the PLL	Depending on the application scenario

Table 3-7 Configuration parameters of the Hi3515 VPLL1

Bit	PLL Pin	Description	Recommended Value
SC_PERCTRL4[31]	dsmpd	0: Decimal frequency-division mode 1: Integer frequency-division mode	Depending on the application scenario
SC_PERCTRL4[30]	bypass	0: no bypass 1: bypass	0b0
SC_PERCTRL4[29:27]	postdiv2	Level-2 output frequency divider of the PLL	Depending on the application scenario
SC_PERCTRL4[26:24]	postdiv1	Level-1 output frequency divider of the PLL	Depending on the application scenario

Bit	PLL Pin	Description	Recommended Value
SC_PERCTRL4[23:0]	frac	Decimal frequency divider of the PLL	Depending on the application scenario
SC_PERCTRL5[22]	reset	PLL reset control 0: reset 1: not reset	0b1
SC_PERCTRL5[21]	pd	PLL power down control 0: disabled 1: enabled	0b1
SC_PERCTRL5[20]	foutvcopd	PLL VCO output power down control 0: disabled 1: enabled	0b0
SC_PERCTRL5[19]	postdivpd	APLL POSTDIV output power down control 0: disabled 1: enabled	0b1
SC_PERCTRL5[18]	vfout4phasepd	PLL FOUT output power down control 0: disabled 1: enabled	0b1
SC_PERCTRL5[17:12]	refdiv	Frequency divider of the PLL reference clock	Depending on the application scenario
SC_PERCTRL5 [11:0]	fbdiv	Integer frequency multiplier of the PLL	Depending on the application scenario

When the clock frequency required by a service is determined, the configured value of the PLL can be calculated accordingly.

Take VPLL0 as an example. Configure VPLL0 to enable it to output FOUTPOSTDIV. Then FOUTPOSTDIV is divided by 2 and output to the VO0 module. Assume that the VO0 module needs a 54 MHz working clock, configure the registers as follows:

- Set postdiv2 to 5 and set postdiv1 to 3, then FOUTVCO = 54253 = 1,620 MHz.
- Set refdiv to 2, then 12 x (fbdiv + frac/2^24) = 1,620 MHz.

Based on the preceding results, fbdiv is 135 and frac is 0.

Take VPLL1 as an example. Configure VPLL1 to enable it to output FOUTPOSTDIV to the VO1 module. Assume that the VO1 module needs a 74.25 MHz working clock, configure the registers as follows:

- Set postdiv2 to 4 and set postdiv1 to 5, then FOUTVCO = 74.2545 = 1,485 MHz.
- Set refdiv to 1, then 24 x (fbdiv + frac/2^24) = 1,485 MHz.

Based on the preceding results, fbdiv is 61 and frac is 14,680,064.

If you want VPLL0 to output a 108 MHz working clock and VPLL1 to output a 65 MHz working clock, configure the registers as follows:

- SC_PERCTRL2 = 0xAB00_0000
- SC_PERCTRL3 = 0x006C_2087
- SC_PERCTRL4 = 0x A300_0000
- 1SC_PERCTRL5 = 0x006C_2041

Frequency Configuration of the ARM/SCLK/HCLK/PCLK

Table 3-8 describes the frequency configuration of the ARM/SCLK/HCLK/PCLK.

Table 3-8 Frequency configuration of the ARM/SCLK/HCLK/PCLK

Signal	Description
sysmode[3:0]	ARM frequency switch. The default value is 0xB that indicates the crystal oscillator mode. The signal is controlled through SC_CTRL [2:0].
sleep_mode	When all the clocks except the SCLK and clocks of the IR and clock reset generation (CRG) module are disabled, the system enters the sleep mode. The signal is controlled through SC_CTRL [2:0].

Table 3-9 describes the mapping between the system controller status and clock switching.

Table 3-9 Mapping between the system controller status and clock switching

Status of the System Controller	Enable Status of the 45 kHz Clock	Enable Status of the 24MHz Crystal Oscillator	Enable Status of the ARMPLL	Status of the System Clock
Normal	Enabled	Enabled	Enabled	The working clocks of the ARM subsystem are derived from the PLL.
Slow	Enabled	Enabled	Disabled	The working clocks of the ARM subsystem are derived from the 24 MHz crystal oscillator.

Status of the System Controller	Enable Status of the 45 kHz Clock	Enable Status of the 24MHz Crystal Oscillator	Enable Status of the ARMPLL	Status of the System Clock
Doze	Enabled	Enabled	Disabled	The working clocks of the ARM subsystem are derived from the 45 kHz clock, which is obtained by dividing the frequency of the 24 MHz crystal oscillator.
Sleep	Enabled	Enabled	Disabled	All the clocks are disabled except that the clocks of the system controller and IR module work at 45 kHz.

Clock Frequency Configuration of Modules

Table 3-10 describes the clock frequency configuration of the MMC module.

Table 3-10 Clock frequency configuration of the MMC module

Signal	Description
mmcclk_sel	Frequency control for the working clock of the MMC module. The signal is controlled through SC_PERCRTL9 bit[21:20].
mmcsap_sel	Normal/reversed phase control for the sampling card data clock of the MMC module. The signal is controlled through SC_PERCRTL bit[22].

Table 3-11 describes the clock frequency configuration of the static memory interface (SMI) module.

Table 3-11 Clock frequency configuration of the SMI module

Signal	Description
ssmcclk_sel	Clock frequency select of the SMI working clock. Through the signal, the frequency of the advanced high-performance bus (AHB) can be divided by 1 or 2. The signal is controlled through SC_PERCRTL9 bit[3].

Table 3-12 describes the clock frequency configuration of the VO module.

Table 3-12 Clock frequency configuration of the VO module

Signal	Description
vo0out_sel	BT656 output clock phase select of the VO0 module, which provides normal or reversed phase output of the working clock of the VO0 module. The signal is controlled through SC_PERCRTL9 bit[18].
vo1out_sel	BT1120 output clock phase clock of the VO1 module, which provides normal or reversed phase output of the working clock of the VO1 module. The signal is controlled through SC_PERCRTL9 bit[19].

Table 3-13 describes the clock frequency configuration of the VI module.

Table 3-13 Clock frequency configuration of the VI module

Signal	Description
vi3div_sel[1:0]	VI3 interface frequency division clock select. Through this signal, the frequency of the VI3 clock can be divided by 1, 2, or 4. The signal is controlled through SC_PERCRTL9 bit[13:12].
vi2div_sel[1:0]	VI2 interface frequency division clock select. Through this signal, the frequency of the VI2 clock can be divided by 1, 2, or 4. The signal is controlled through SC_PERCRTL9 bit[11:10].
vi1div_sel[1:0]	VI1 interface frequency division clock select. Through this signal, the frequency of the VI1 clock can be divided by 1, 2, or 4. The signal is controlled through SC_PERCRTL9 bit[9:8].
vi0div_sel[1:0]	VI0 interface frequency division clock select. Through this signal, the frequency of the VI0 clock can be divided by 1, 2, or 4. The signal is controlled through SC_PERCRTL9 bit[7:6].
vi3_vi2_sel	VI3 interface clock select. The signal is controlled through SC_PERCRTL9 bit[5].
vi1_vi0_sel	VI1 interface clock select. The signal is controlled through SC_PERCRTL9 bit[4].

Table 3-14 describes the clock frequency configuration of SIO0/SIO1.

Table 3-14 Clock frequency configuration of SIO0/SIO1

Signal	Description
sio0clk_sel[23:0]	Frequency divider select of the output interface master clock SIO_MCLK of the SIO0 module. The signal is controlled through SC_PERCRTL13 bit[23:0].

Signal	Description
sio0_bclk_sel[3:0]	Frequency configuration of the bit stream clock SIO_XCK (BCLK) of the SIO1 module in master mode. The signal is controlled through SC_PERCRTL13 bit[27:24].
sio0_lrclk_sel[3:0]	Frequency configuration of the sampling rate output clock SIO_RFS/SIO_XFS (FSCLK) of the SIO0 module in master mode. The signal is controlled through SC_PERCRTL13 bit[31:28].
sio1clk_sel[23:0]	Frequency divider select of the output interface master clock SIO_MCLK of the SIO0 module. The signal is controlled through SC_PERCRTL14 bit[23:0].
sio1_bclk_sel[3:0]	Frequency configuration of the bit stream clock SIO_XCK (BCLK) of the SIO1 module in master mode. The signal is controlled through SC_PERCRTL14 bit[27:24].
sio1_lrclk_sel[3:0]	Frequency configuration of the sampling rate output clock SIO_RFS/SIO_XFS (FSCLK) of the SIO1 module in master mode. The signal is controlled through SC_PERCRTL14 bit[31:28].
sio0_blk_edge	Bit stream clock normal/reverse phase control of the SIO0 module. The signal can be controlled through SC_PERCRTL16 bit[8].
sio1_blk_edge	Bit stream clock normal/reverse phase control of the SIO1 module. The signal can be controlled through SC_PERCRTL16 bit[9].
aclkout_sel	ACKOUT output clock. The signal is controlled through SC_PERCRTL9 bit[25].

The frequency FSCLK of the sampling rate clock is specified in common application scenarios; however, the frequencies of the bit clock BCLK and the master clock MCLK are variable multiples of FSCLK. The following example shows how to configure the clock frequencies.

Assume that the frequency of the clock source ARMPLL is set to 540 MHz and the required working clock frequencies are as follows: FSCLK = 48kHz, MCLK = 256FSCLK = 12.288MHz, and BCLK = 16FSCLK = 384KHz. The following configuration is performed:

- The frequency division ratio N of ARMPLL to MCLK is 12.288/540, then sioclk_sel[23:0] = N x 2^{27}. According to the result 3,054,199 obtained by rounding N x 2^{27} to the nearest integer, set sioclk_sel to 0x002E_9A77. In this way, the correct MCLK frequency is obtained.
- The frequency of BCLK is generated by dividing that of BCLK. The frequency division ratio of BCLK to MCLK is 16/256 = 1/16. According to the mapping in Table 3-14, set siobclk_sel[2:0] to 0b100 (corresponding to the frequency divider 16). In this way, the correct frequency of BCLK is obtained.
- The frequency of FSCLK is generated by dividing that of BCLK and their frequency division ratio is 1/16. According to the mapping in Table 3-14, set siolrclk_sel to 0b011 (corresponding to the frequency divider 16). In this way, the correct frequency of FSCLK is obtained.

Clock Gating Configuration

For details about the clock gating configuration, see the description of the system controller SC_PEREN.

Precautions

Take the following precautions when configuring clocks:

- By default, the ARM working clock is in crystal oscillator mode after power-on. That is, the crystal oscillator clock input by the XIN24 pin is selected.
- If the frequency of the PLL is changed, the PLL can generate a clock with a stable frequency until 0.1 ms later. The frequency of the PLL can be changed only when the system runs in slow mode.
- When the PLL output clock is not stable, the system cannot be switched to the PLL mode.

3.3 Processor and Memory Address Mapping

3.3.1 Processor

The Hi3515 has an embedded processor ARM926EJ-S. The features of ARM926EJ-S are as follows:

- Adopts the 32-bit ARM v5TEJ and 5-stage pipeline to be compatible with the 32-bit ARM and the 16-bit thumb instruction sets.
- Supports the embedded enhanced digital signal processing (DSP) instructions.
- Supports Java.
- Provides the independent 16 KB instruction cache, 16 KB data cache, and 4-way set associative cache. The cache line size is 32 bytes. The data cache supports configurable write-back and write-through operations.
- Provides the caches that supports configurable pseudo-random or round-robin replacement algorithm.
- Provides independent instructions and data bus interfaces. The ratio of the operating frequency of the bus to that of the system clock of ARM926EJ-S can be set to 1:1 or 1:2.
- Includes an MMU that supports multiple open operating systems, such as VxWorks, Linux, WindowCE, and PalmOS.
- Provides an independent 2-KB instruction tightly-coupled memory (ITCM).
- Adopts the little endian mode.
- Supports fast interrupt requests (FIQs) and interrupt requests (IRQs).
- Supports the joint test action group (JTAG) debugging interface.
- Supports dynamic and static power management.

3.3.2 Memory Address Mapping

The Hi3515 uses 32-bit address bus and supports 4 GB addressable space. The Hi3515 can boot in different modes by configuring the BOOTSEL pin:

- 00: boot from the NOR flash
- 00: boot from the NAND flash

Booting from the NOR Flash

The Hi3515 can boot from the NOR flash only after the BOOTSEL pin is configured properly. In this boot mode, the externally-connected memory is the asynchronous NOR flash. The Hi3515 supports the 8-bit NOR flash only.

Figure 3-3 shows the address space mapping of the Hi3515 when the it boots from the NOR flash.

Figure 3-3 Address space mapping when the Hi3515 from the NOR flash

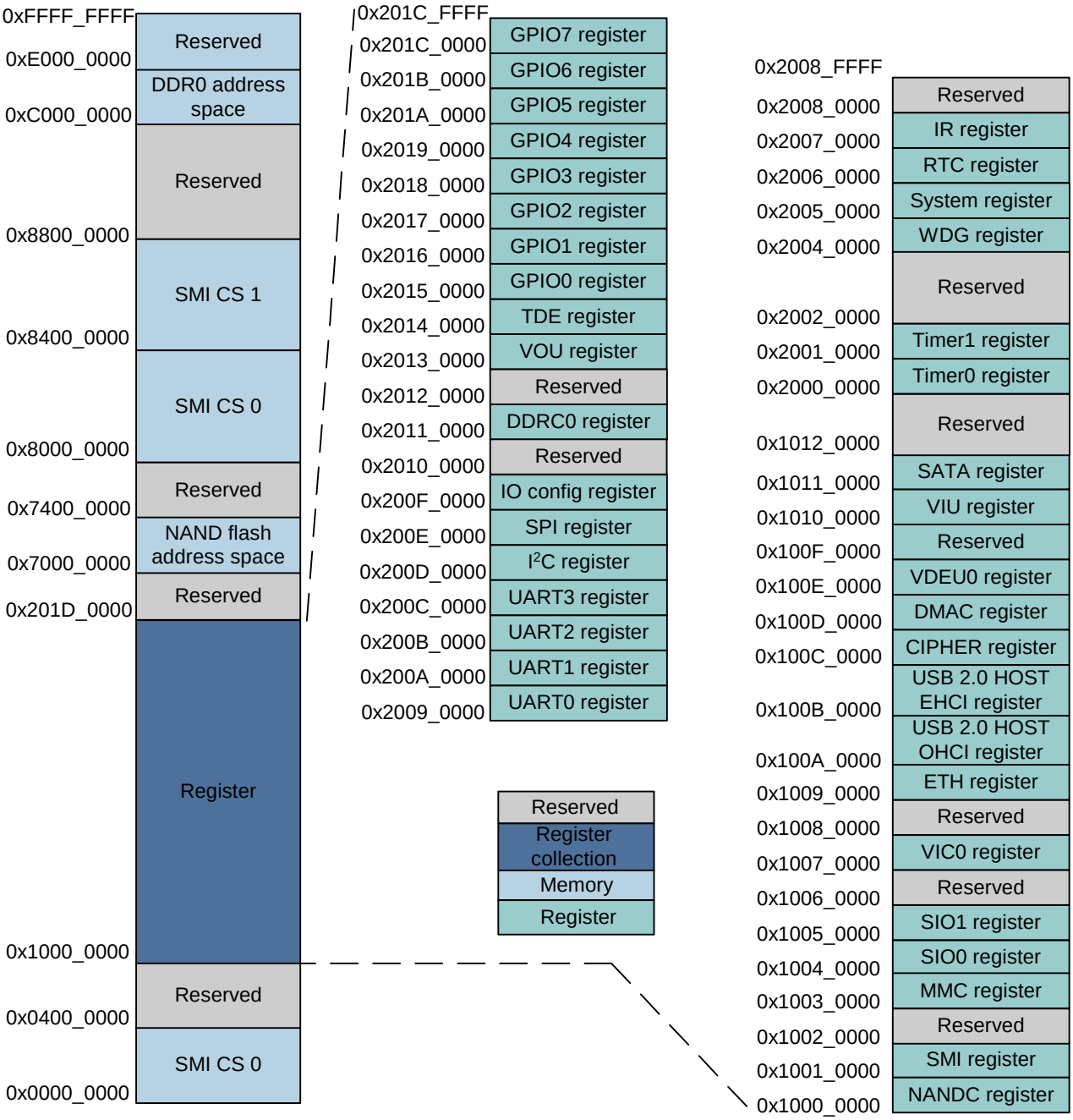
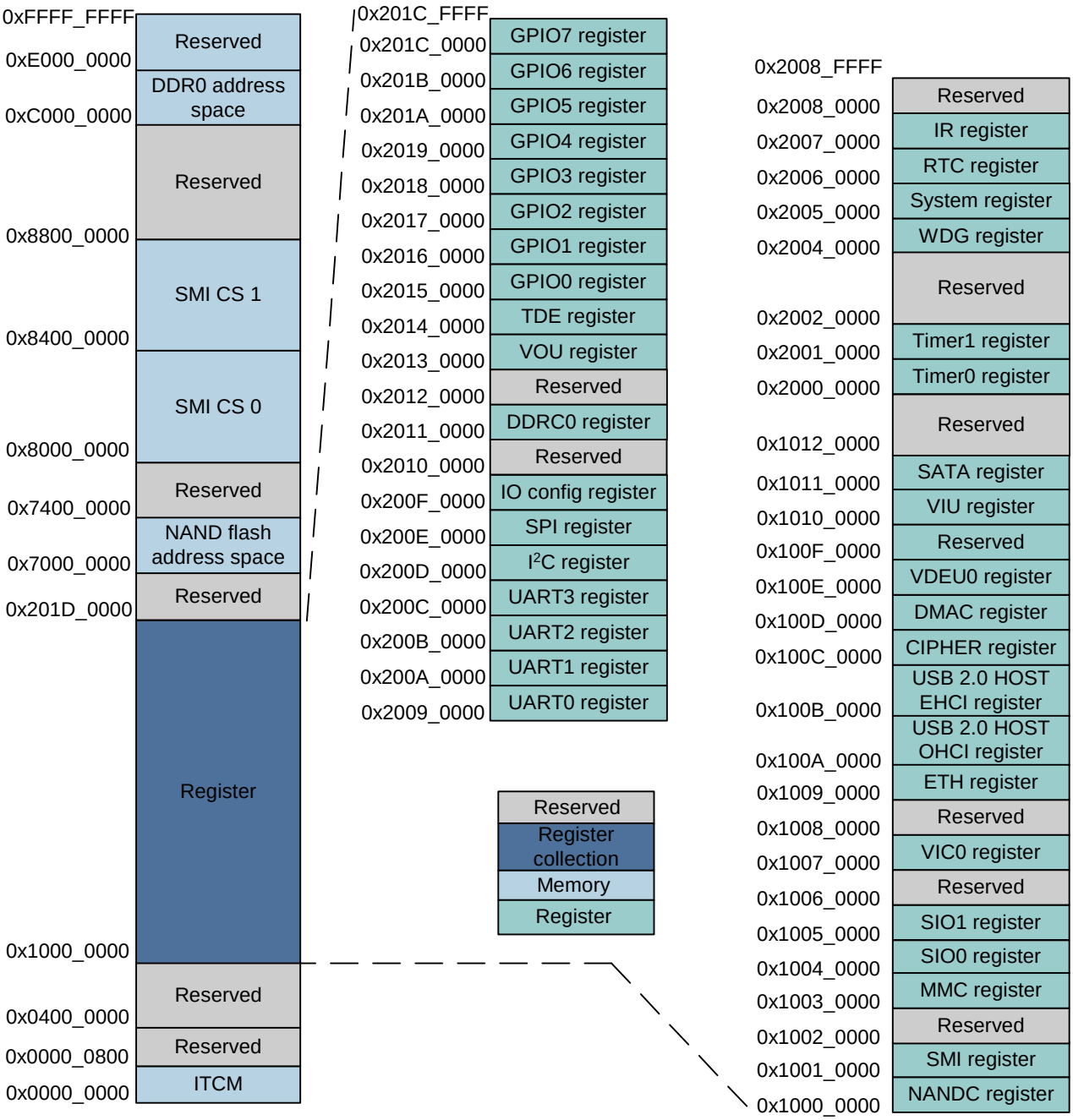


Figure 3-4 shows the address distribution after the remapping is cleared during the booting from the NOR flash.

CAUTION

- The system actually provides 2 KB ITCM address space only. Hence, ensure that the size of the program and data in the ITCM is equal to or less than 2 KB.
- If you want to use the ITCM, you must enable the ITCM and configure the ITCM size by setting the register of the system control coprocessor CP15 of ARM. In addition, you need to set the C9 register to 0xD by using the MCR instruction.

Figure 3-4 Address distribution after the remapping is cleared during the booting from the NOR flash.



Booting from the NAND Flash

The Hi3515 can boot from the NAND flash only after the BOOTSEL pin is configured properly. In this boot mode, the externally-connected memory is the NAND flash. The Hi3515 supports the 8-bit NAND flash only.

Figure 3-5 shows the address space mapping of the Hi3515 when it boots from the NAND flash.

Figure 3-5 Address space mapping when the Hi3515 boots from the NAND flash

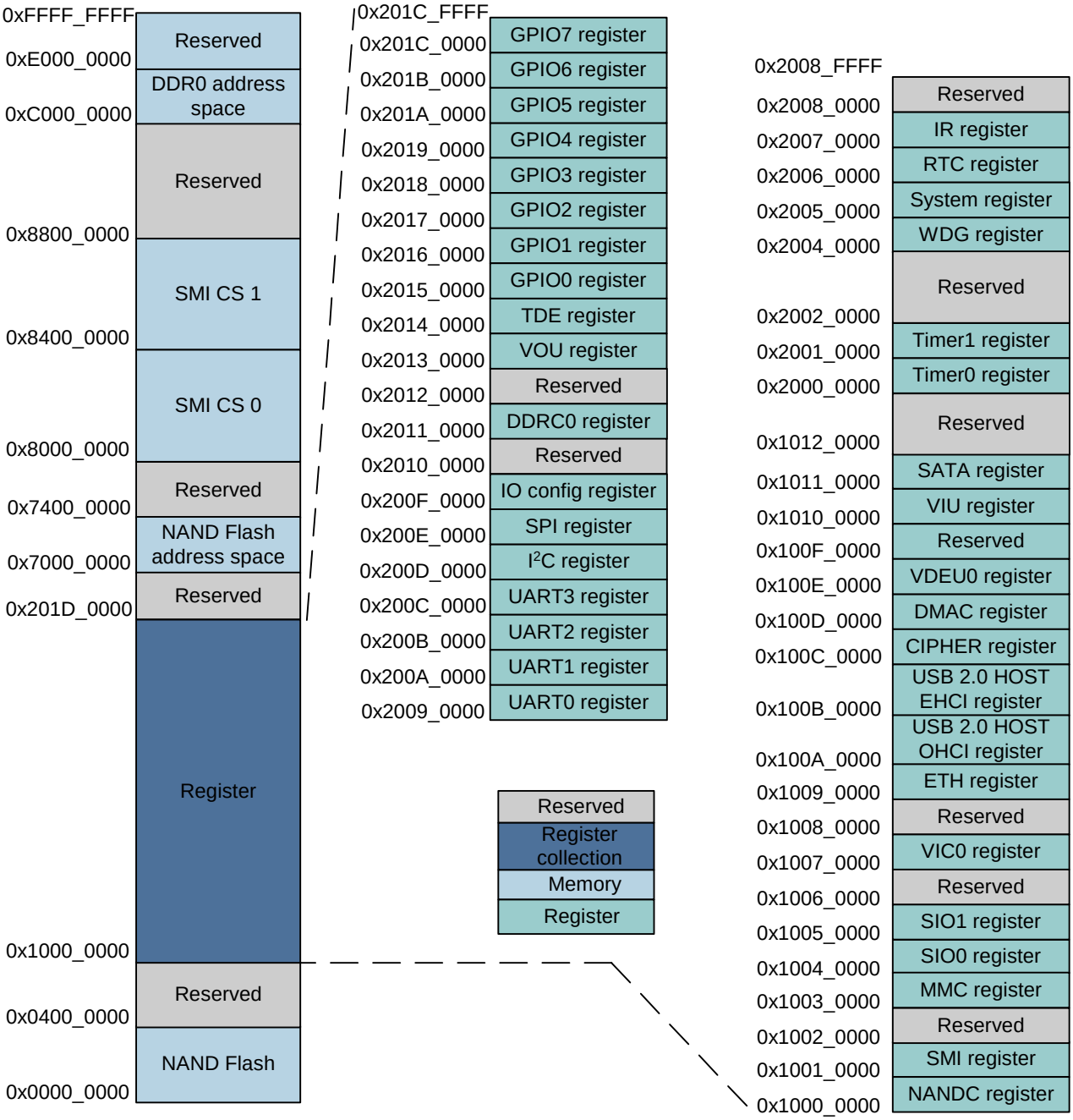
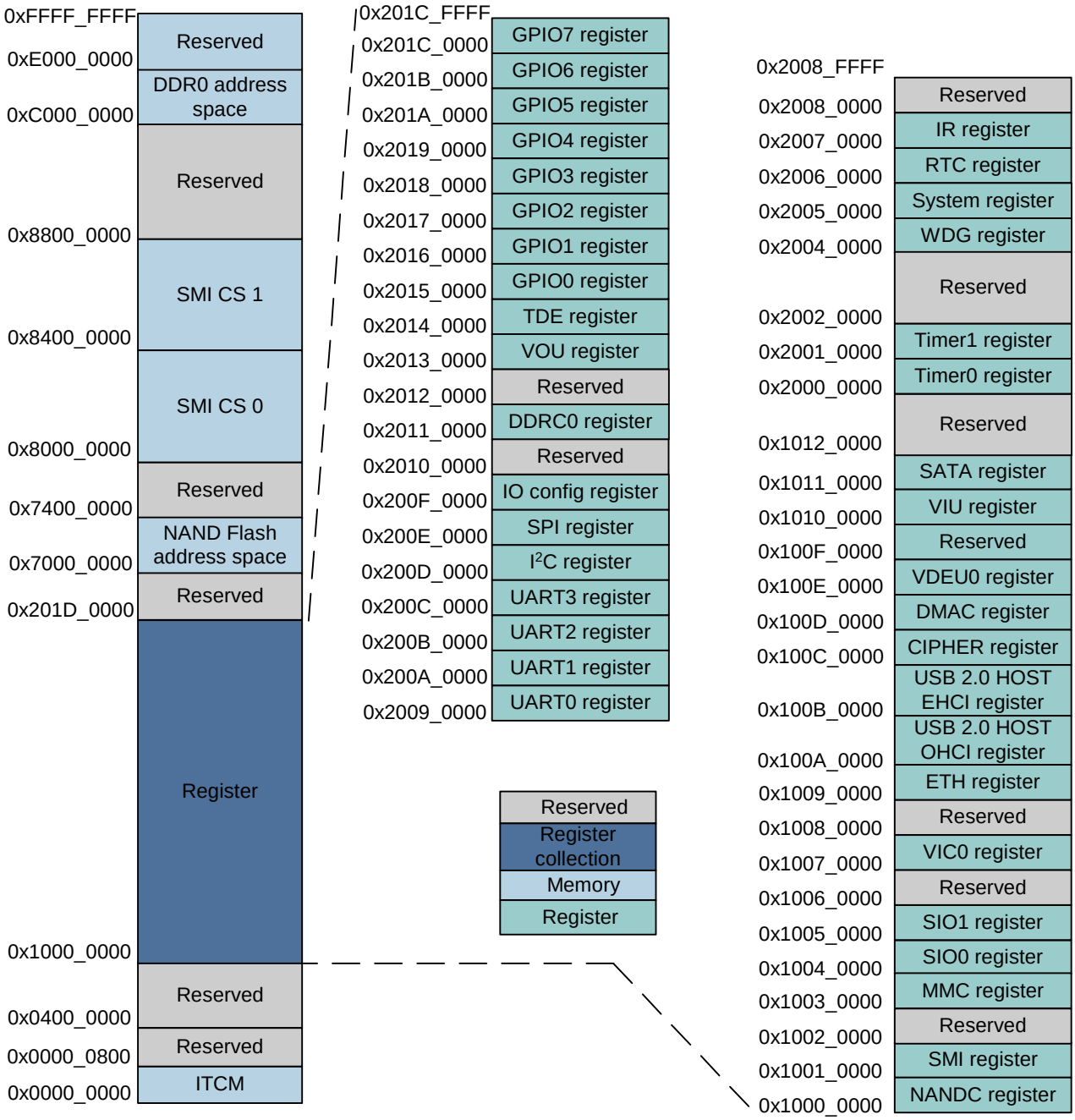


Figure 3-6 shows the address distribution after the remapping is cleared during the booting from the NAND flash.

Figure 3-6 Address distribution after the remapping is cleared during the booting from the NAND flash



Address Space List

Table 3-15 lists the address spaces of the Hi3515.

Table 3-15 Address spaces of the Hi3515

Start Address	End Address	Function	Capacity	Description
0xE000_0000	0xFFFF_FFFF	Reserved.	512 MB	–
0xC000_0000	0xDFFF_FFFF	DDR0	512 MB	Addressing space of the dynamic memory (128 bits)
0xB000_0000	0xBFFF_FFFF	Reserved.	256 MB	–
0x8800_0000	0xAFFF_FFF	Reserved.	640 MB	–
0x8400_0000	0x87FF_FFFF	SSMC CS1	64 MB	–
0x8000_0000	0x83FF_FFFF	SSMC CS0	64 MB	–
0x7400_0000	0x7FFF_FFFF	Reserved.	64 MB	–
0x7000_0000	0x73FF_FFFF	Storage space of the NAND flash	64 MB	–
0x201D_0000	0x6FFF_FFFF	Reserved.	1278 MB	–
0x201C_0000	0x201C_FFFF	GPIO7 register	64 KB	–
0x201B_0000	0x201B_FFFF	GPIO6 register	64 KB	–
0x201A_0000	0x201A_FFFF	GPIO5 register	64 KB	–
0x2019_0000	0x2019_FFFF	GPIO4 register	64 KB	–
0x2018_0000	0x2018_FFFF	GPIO3 register	64 KB	–
0x2017_0000	0x2017_FFFF	GPIO2 register	64 KB	–
0x2016_0000	0x2016_FFFF	GPIO1 register	64 KB	–
0x2015_0000	0x2015_FFFF	GPIO0 register	64 KB	–
0x2014_0000	0x2014_FFFF	TDE register	64 KB	–
0x2013_0000	0x2013_FFFF	VOU register	64 KB	–
0x2012_0000	0x2012_FFFF	Reserved.	64 KB	–
0x2011_0000	0x2011_FFFF	DDRC0 register	64 KB	–
0x2010_0000	0x2010_FFFF	Reserved.	64 KB	–
0x200F_0000	0x200F_FFFF	IO config register	64 KB	–
0x200E_0000	0x200E_FFFF	SPI register	64 KB	–
0x200D_0000	0x200D_FFFF	I^2C register	64 KB	–
0x200C_0000	0x200C_FFFF	UART3 register	64 KB	–
0x200B_0000	0x200B_FFFF	UART2 register	64 KB	–
0x200A_0000	0x200A_FFFF	UART1 register	64 KB	–

Start Address	End Address	Function	Capacity	Description
0x2009_0000	0x2009_FFFF	UART0 register	64 KB	–
0x2008_0000	0x2008_FFFF	Reserved.	64 KB	–
0x2007_0000	0x2007_FFFF	IR register	64 KB	–
0x2006_0000	0x2006_FFFF	RTC register	64 KB	–
0x2005_0000	0x2005_FFFF	System control register	64 KB	–
0x2004_0000	0x2004_FFFF	WDG register	64 KB	–
0x2002_0000	0x2003_FFFF	Reserved.	128 KB	–
0x2001_0000	0x2001_FFFF	Timer1 register	64 KB	–
0x2000_0000	0x2000_FFFF	Timer0 register	64 KB	–
0x1012_0000	0x1FFF_FFFF	Reserved.	255 MB	–
0x1011_0000	0x1011_FFFF	SATA register	64 KB	–
0x1010_0000	0x1010_FFFF	VIU register	64 KB	–
0x100F_0000	0x100F_FFFF	Reserved.	64 KB	–
0x100E_0000	0x100E_FFFF	VEDU0 register	64 KB	–
0x100D_0000	0x100D_FFFF	DMAC register	64 KB	–
0x100C_0000	0x100C_FFFF	CIPHER register	64 KB	–
0x100B_0000	0x100B_FFFF	USB 2.0 HOST EHCI register	64 KB	–
0x100A_0000	0x100A_FFFF	USB 2.0 HOST OHCI register	64 KB	–
0x1009_0000	0x1009_FFFF	ETH register	64 KB	–
0x1008_0000	0x1008_FFFF	Reserved.	64 KB	–
0x1007_0000	0x1007_FFFF	VIC0 register	64 KB	–
0x1006_0000	0x1006_FFFF	Reserved.	64 KB	–
0x1005_0000	0x1005_FFFF	SIO1 register	64 KB	–
0x1004_0000	0x1004_FFFF	SIO0 register	64 KB	–
0x1003_0000	0x1003_FFFF	SDIO register	64 KB	–
0x1002_0000	0x1002_FFFF	Reserved.	64 KB	–
0x1001_0000	0x1001_FFFF	SSMC register	64 KB	–
0x1000_0000	0x1000_FFFF	NANDC register	64 KB	–
0x0400_0000	0x0FFF_FFFF	Reserved.	192 MB	–

Start Address	End Address	Function	Capacity	Description
0x0000_0000	0x03FF_FFFF	Memory selected for address remapping	64 MB	The component corresponding to the current storage space is configured based on the BOOTSEL pin during reset. In general, the value of the storage space is set to the value of the ITCM. after the address remapping is cleared. The size of the ITCM is 2 KB, and its address space ranges from 0x0000_0000 to 0x0000_08000.

NOTE

- DDR = double-data rate
- SSMC = static memory interface controller
- GPIO = general purpose input/output
- TDE = two-dimensional engine
- VOU = video output unit
- DDRC = double-data rate controller
- SPI = serial peripheral interface
- I2C = inter-integrated circuit
- UART = universal asynchronous receiver transmitter
- WDG = watchdog
- SATA = serial advanced technology attachment
- VIU = video input video
- VEDU = video encoding decoding unit
- DMAC = direct memory access controller
- EHCI = enhanced host controller interface
- OHCI = open host controller interface
- VIC = vector interrupt controller
- SDIO = secure digital input/output
- NANDC = NAND flash Controller

3.4 Interrupt System

3.4.1 Overview

The interrupt system (INT) of the Hi3515 provides the interrupt management function.

3.4.2 Features

The INT has the following features:

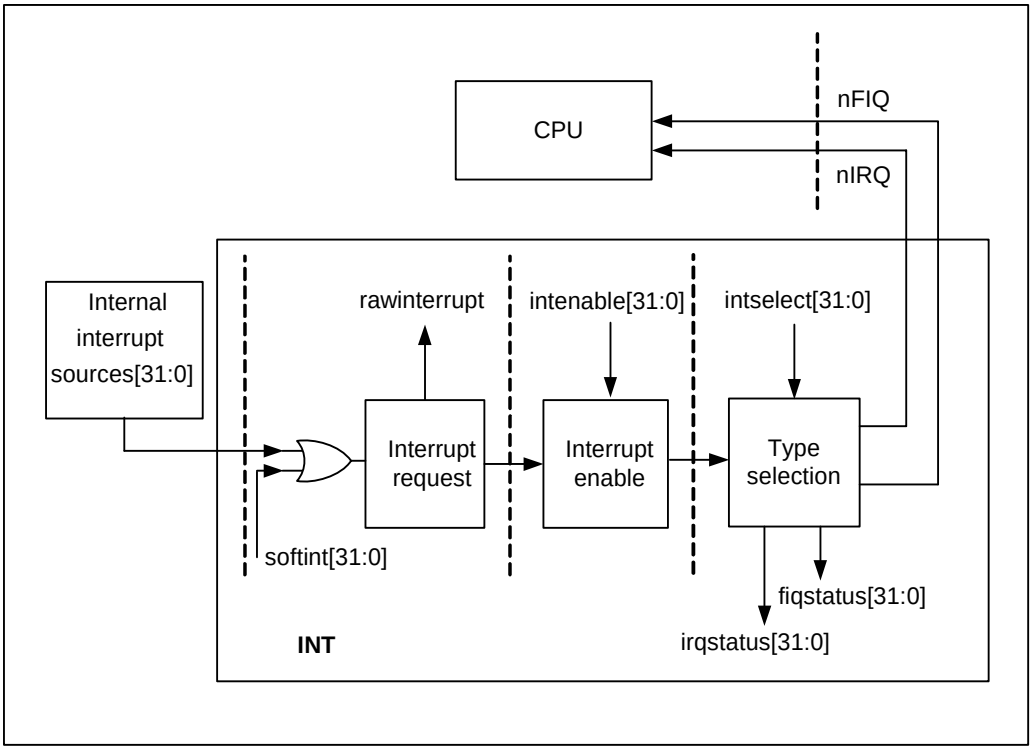
- Supports 32 interrupt sources that are triggered by high level.
- Generates interrupt requests when the INT working clock is disabled.
- Supports configurable interrupt types. Each interrupt source is set to IRQ or FIQ as required.
- Masks interrupting sources.
- Queries the status of raw interrupt sources and masked interrupt sources.
- Supports the access protection function. When the access protection function is enabled, the CPU can access the INT in privileged mode only.

3.4.3 Function Description

Functional Block Diagram

Figure 3-7 shows the functional block diagram of the INT.

Figure 3-7 Functional block diagram of the INT



NOTE

In Figure 3-7, the letter *n* in nFIQ and nIRQ indicates active low

The CPU can access the function registers of the INT through the internal bus. For example:

- Enabling or masking the interrupt request of each interrupt source by configuring INT_INTENABLE
- Selecting the interrupt request type of each interrupt source by configuring INT_INTSELECT. Generally, up to one interrupt request is set to FIQ.
- Generating and clearing software interrupt requests by configuring INT_SOFTINT and INT_SOFTINTCLEAR respectively
- Querying the status of interrupt sources by configuring INT_IRQSTATUS and INT_FIQSTATUS when interrupt requests are generated
- Querying the raw interrupt requests that are generated by interrupt sources and software interrupts by configuring INT_RAWINTR

Interrupt Processing

When an interrupt occurs, the interrupt is processed as follows:

Step 1 Switch the ARM processor to the IRQ or FIQ interrupt mode.

Step 2 Jump to the primary interrupt handler and push the stack.

Step 3 Start the system.

Step 4 Query INT_IRQSTATUS or INT_FIQSTATUS to determine the interrupt source. If there are multiple active interrupt sources, compare the interrupt priorities.

Step 5 Jump to the interrupt service routine (ISR) corresponding to each interrupt source and configure INT_INTENABLE to mask the interrupt that is being processed. If necessary, configure INT_INTENABLE to enable corresponding interrupts.

Step 6 Run the ISR.

Step 7 Clear the current interrupt source. If it is a software interrupt, configure the corresponding bit of INT_SOFTINTCLEAR.

Step 8 Enable the IRQ and FIQ mask bits of ARM CPSR.

Step 9 Pop the stack and return from the interrupt.

----End

Interrupt Mapping

Table 3-16 lists the interrupt mapping of the INT.

Table 3-16 Interrupt mapping of the INT

Interrupt No.	Interrupt Source
0	Watchdog interrupt
1	Global software interrupt
2	COMMRX interrupt

Interrupt No.	Interrupt Source
3	COMMTX interrupt
4	Dual-timer01 interrupt
5	Dual-timer23 interrupt
6	GPIO0 interrupt
7	GPIO1 interrupt
8	Combined interrupt of GPIO2, GPIO3, GPIO4, and GPIO5
9	IR interrupt
10	RTC interrupt
11	SPI interrupt
12	UART0 or UART1 interrupt
13	UART2 or UART3 interrupt
14	ETH interrupt
15	DMAC interrupt
16	I^2C interrupt
17	VIU interrupt
18	TDE interrupt
19	VOU interrupt
20	VEDU interrupt
21	Reserved.
22	USB 2.0 HOST OHCI interrupt
23	USB 2.0 HOST EHCI interrupt
24	SDIO interrupt
25	SIO0 interrupt
26	SIO1 interrupt
27	Reserved.
28	CIPHER interrupt
29	SATA interrupt
30	NANDC interrupt
31	Reserved.

NOTE
The COMMRX and COMMTX interrupts are used for CPU debugging.

3.4.4 Register Summary

Table 3-17 lists the INT registers.

Table 3-17 Summary of the INT registers (based address: 0x1007_0000)

Offset Address	Register	Description	Page
0x000	INT_IRQSTATUS	IRQ interrupt status register	3-27
0x004	INT_FIQSTATUS	FIQ interrupt status register	3-28
0x008	INT_RAWINTR	Raw interrupt status register	3-28
0x00C	INT_INTSELECT	Interrupt source select register	3-28
0x010	INT_INTENABLE	Interrupt enable register	3-29
0x014	INT_INTENCLEAR	Interrupt enable clear register	3-29
0x018	INT_SOFTINT	Software interrupt register	3-30
0x01C	INT_SOFTINTCLEAR	Software interrupt clear register	3-30
0x020	INT_PROTECTION	Protection enable register	3-31

3.4.5 Register Description

INT_IRQSTATUS

INT_IRQSTATUS is the IRQ interrupt status register. Its 32 bits map to 32 interrupt sources. For details, see Table 3-16 .

Offset Address	Register Name	Total Reset Value
0x000	INT_IRQSTATUS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	irqstatus	Status of the IRQ interrupt source. 0: No interrupt is generated. 1: An IRQ interrupt is generated and sent to the processor.

INT_FIQSTATUS

INT_FIQSTATUS is the FIQ interrupt status register. The 32 bits of INT_INTENABLE map to 32 interrupt sources.

Offset Address	Register Name	Total Reset Value
0x004	INT_FIQSTATUS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	fiqstatus	Status of the FIQ interrupt source. 0: No interrupt is generated. 1: An FIQ interrupt is generated and sent to the processor.

INT_RAWINTR

INT_RAWINTR is the raw interrupt status register. It shows the status of raw interrupt requests and the status of the software interrupts that are generated by configuring INT_SOFTINT. The 32 bits of INT_RAWINTR map to 32 interrupt sources.

Offset Address	Register Name	Total Reset Value
0x008	INT_RAWINTR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	rawinterrupt	Status of the interrupt requests of raw interrupt sources. 0: No interrupt is generated. 1: An interrupt is generated.

INT_INTSELECT

INT_INTSELECT is the interrupt source select register. It determines whether the selected interrupt sources generate IRQ interrupts or FIQ interrupts. The 32 bits of INT_INTSELECT map to 32 interrupt sources.

Offset Address	Register Name	Total Reset Value
0x00C	INT_INTSELECT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	intselect	Interrupt request type of interrupt sources. 0: IRQ interrupt 1: FIQ interrupt

INT_INTENABLE

INT_INTENABLE is the interrupt enable register. It is used to enable interrupt request lines. After reset, the value of INT_INTENABLE is changed to 0x0000_0000. As a result, all interrupt sources are masked. The 32 bits of INT_INTENABLE map to 32 interrupt sources.

Offset Address	Register Name	Total Reset Value
0x010	INT_INTENABLE	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	intenable	Returned mask status of each interrupt source when this register is read. 0: masked 1: not masked Interrupt source is enabled bit by bit when this register is written. 0: The current value of the corresponding bit is not affected. 1: The corresponding bit is set to 1 to enable the corresponding interrupt request.

INT_INTENCLEAR

INT_INTENCLEAR is the interrupt enable clear register. It is used to clear the corresponding bits of INT_INTENABLE. INT_INTENCLEAR is write-only and has no default reset value.

Offset Address	Register Name	Total Reset Value
0x014	INT_INTENCLEAR	-

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

Bits	Access	Name	Description
[31:0]	WO	intenableclear	Mask of the interrupt source corresponding to INT_INTENABLE. 0: The current value of the corresponding bit of INT_INTENABLE is not affected. 1: The corresponding bit of INT_INTENABLE is cleared and the corresponding interrupt request is masked.

INT_SOFTINT

INT_SOFTINT is the software interrupt register. Through software, it controls whether to generate interrupts by the interrupt source input lines. The software interrupts can be cleared by writing INT_SOFTINTCLEAR. The clear operation is performed after the ISR is executed.

Offset Address	Register Name	Total Reset Value
0x018	INT_SOFTINT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	softint	Generates a raw software interrupt on a specified interrupt source. 0: The current value of the corresponding bit is not affected. 1: The corresponding bit is set to 1 and a software interrupt request is generated.

INT_SOFTINTCLEAR

INT_SOFTINTCLEAR is the software interrupt clear register. It is used to clear the corresponding bit of INT_SOFTINT. INT_SOFTINTCLEAR is write-only and has no default reset value.

Offset Address	Register Name	Total Reset Value
0x01C	INT_SOFTINTCLEAR	-

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

Bits	Access	Name	Description
[31:0]	WO	softintclear	Mask of the interrupt request corresponding to INT_SOFTINT. 00: The corresponding bit of INT_SOFTINT is not affected. 1: The corresponding bit of INT_SOFTINT is cleared and the corresponding interrupt request is masked.

INT_PROTECTION

INT_PROTECTION is the protection enable register It is used to enable or disable the access to the protected registers.

CAUTION

- After reset, this register is cleared and the registers of the INT can be accessed in both user mode and privileged mode.
- When the CPU cannot generate the correct protection information (HPROT), the registers of the INT can be accessed in user mode after INT_PROTECTION is reset.

Offset Address	Register Name	Total Reset Value
0x020	INT_PROTECTION	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RW	protection	Register access protection enable. 0: disabled. The registers of the INT can be accessed both in user mode and privileged mode. 1: enabled. The registers of the INT can be access in privileged mode only.

3.5 Direct Memory Access Controller

3.5.1 Overview

DMA refers to an operating mode in which data is transferred through input/output (I/O) exchange by hardware only. In this mode, the DMAC directly transfers data between a memory and a peripheral, between peripherals, and between memories. This avoids the processor intervention and reduces the interrupt processing overhead of the processor. The DMA mode is usually used to transmit data blocks at high speed. After receiving a DMA transfer request, the DMAC enables the master bus controller based on the channel configuration, sends address and control signals to memories and peripherals, counts the transferred data segments, and reports the end of data transfer or errors to the CPU in interrupt mode.

3.5.2 Features

The DMAC has the following features:

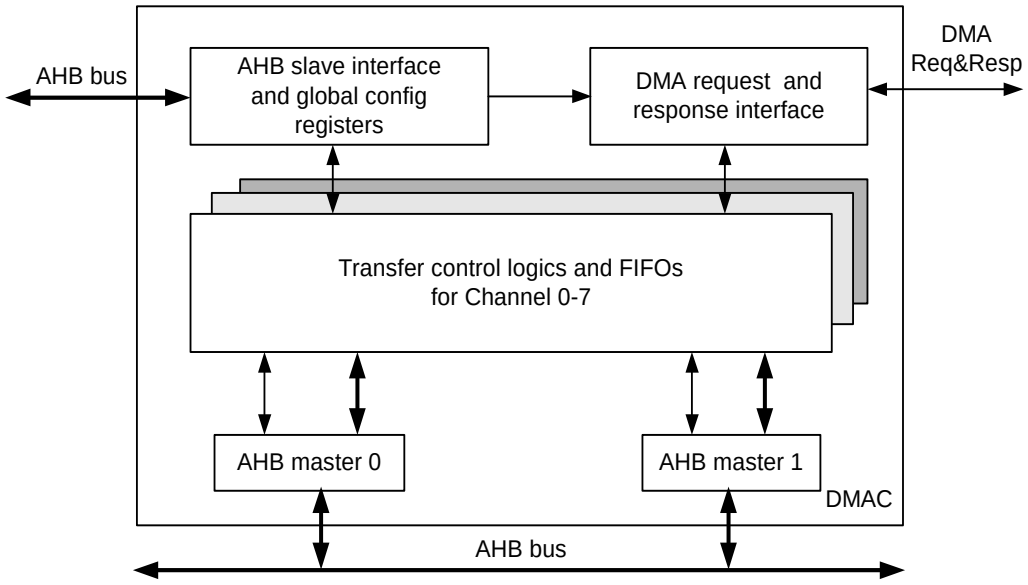
- Transfers data in 8-bit, 16-bit, or 32-bit mode.
- Provides eight DMA channels. Each channel can be configured to support unidirectional transfer.
- Provides eight fixed DMA channel priorities. DMA channel 0 has the highest priority; whereas channel 7 has the lowest priority. When the DMA requests from two peripherals are valid simultaneously, the channel with the higher priority starts data transfer first.
- Includes one 4 x 32-bit first in first out (FIFO) in each DMAC channel (channel 0 to channel 5) and one 16 x 32-bit FIFO in each DMAC channel (channel 6 to channel 7).
- Provides two 32-bit master bus interfaces for data transfer.
- Supports two types of DMA requests: single transfer and burst transfer.
- Provides 16 groups of DMA request inputs. These requests can be configured as source requests or target requests.
- Supports the DMA requests controlled through software.
- Supports the DMA burst size configured by programming.
- Supports the source address and the target address that are automatically incremented or not during DMA transfer
- Supports four data transfer directions:
  - Memory to peripheral
  - Memory to memory
  - Peripheral to memory
  - Peripheral to peripheral
- Supports DMA transfer with the linked list.
- Supports the DMAC flow control.
- Providing two maskable interrupt outputs that are reported to the master ARM and slave ARM respectively, querying the status of the raw/masked DMA error interrupt and DMA terminal count interrupt, and querying the status of the combination of the two interrupts.
- Controls the power consumption by disabling the DMAC and supports DMAC clock gating.

3.5.3 Function Description

3.5.3.1 Functional Block Diagram

Figure 3-8 shows the functional block diagram of the DMAC.

Figure 3-8 Functional block diagram of the DMAC



Each DMAC channel involves a group of transfer control logics and one FIFO. This group of transfer control logics perform the following operations automatically:

Step 1 Read data from a specified source address.

Step 2 Buffer the data to the FIFO.

Step 3 Take data out from the FIFO.

Step 4 Write the data to a specified target address.

----End

3.5.3.2 Workflow

The workflow of the DMAC is as follows:

Step 1 The software selects one DMAC channel for DMA transfer, configures the following items for this channel, and then enables this channel:

- Source address
- Target address
- Head pointer of the linked list
- Amount of the transferred data
- Peripheral request signal line numbers at the source and target sides

- Masters at the source and target sides

After the channel is enabled, the DMAC hardware starts to check the activities on the DMA request signal lines of the source peripheral and target device connected to this channel.

Step 2 The source device sends a DMA request to the DMAC. If the source device is a memory, the DMAC considers that the DMA request is always valid by default.

Step 3 The DMAC channel responds to the DMA request of the source device. The DMAC reads data from the source device and stores it in the internal FIFO of the channel.

Step 4 The target device sends a DMA request to the DMAC. If the target device is a memory, the DMAC considers that the DMA request is always valid by default.

Step 5 The DMAC channel responds to the DMA request of the target device. The DMAC takes out data from the internal FIFO of the channel and writes it to the target device.

Step 6 Step 2, Step 3 and Step 4, Step 5 may be performed concurrently, because the source and target devices may send DMA requests to the DMAC at the same time.

When the FIFO overrun or underrun of the DMA channel occurs, the DMAC blocks the DMA requests of the source device or target device until the FIFO is not full or empty.

When the DMAC interacts with the source device and target device for several times, Step 2 to Step 5 are performed repeatedly until the specified data is transferred and a maskable terminal count interrupt is sent.

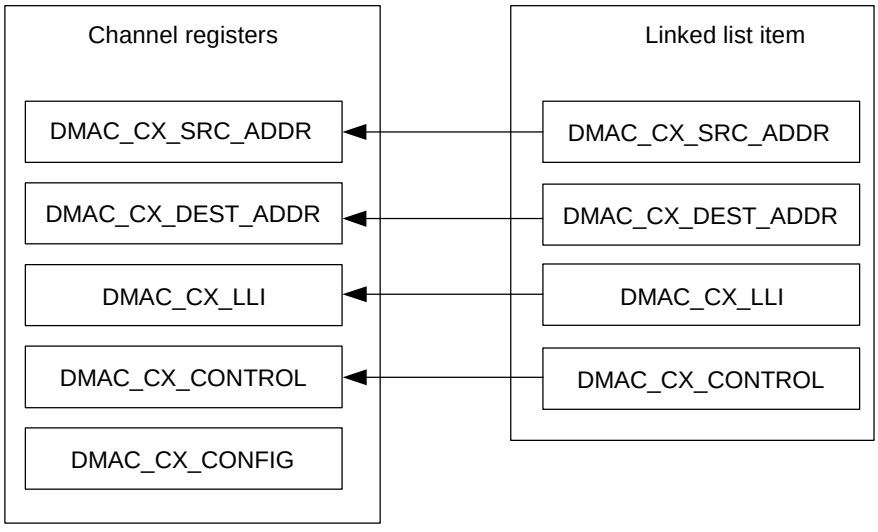
If the value of DMAC_CX_LLI is not 0, read linked list item (LLI) nodes by considering the register value as an address, load the read values to DMAC_CX_SRC_ADDR, DMAC_CX_DEST_ADDR, DMAC_CX_LLI, and DMAC_CX_CONTROL in sequence (see Figure 3-9), and then go to Step 2.

If the value of DMAC_CX_LLI is 0, the current DMA transfer is stopped. In this case, the channel is disabled automatically and the transfer ends.

----End

Figure 3-9 shows how to update channel registers through the LLI.

Figure 3-9 Updating channel registers through the LLI



3.5.3.3 Connections Between the DMA and Peripherals

The peripherals initiate data transfer by sending DMA request signals to the DMAC.

DMA Request Signals

The DMAC provides two types of DMA request signals for each peripheral:

- DMACBREQ

  Burst transfer request signal. It triggers a burst transfer. The burst size is preset.
- DMACSREQ

  Single transfer request signal. It triggers a single transfer. That is, the DMAC reads a data segment from a peripheral or writes a data segment to a peripheral.

Request Clear Signal

The DMAC provides a request clear signal DMACLR.

This signal is sent to each peripheral by the DMAC and is used to respond to the DMA request signal of each peripheral.

DMAC Request Signals

Table 3-18 shows the hardware request signals of the DMAC.

Table 3-18 Hardware request signals of the DMAC

DMAC Hardware Request Signal No.	Description
0	SIO0 receive (RX) channel
1	SIO0 transmit (TX) channel
2	SIO1 RX channel
3	SIO1 TX channel
4	UART RX channel
5	UART TX channel
6	SPI RX channel
7	SPI TX channel
8	MMC read channel
9	MMC write 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

DMAC Hardware Request Signal No.	Description
14	DMA request of the UART2 RX channel
15	DMA request of the UART2 TX channel

The source and target requests of each DMA channel are configured through the software. For example, DMA request 2 is the request of the RX channel of SIO1. If you want to transmit the received data of SIO1 through channel 3, you must configure DMA request 2 as the source request of channel 3.

There is no DMA request signal for memories. When a memory is used, the DMAC considers that the DMA request of the memory is always valid by default. For DMAC channel 6 and channel 7, an idle cycle is inserted after each bus operation. In this way, the master with a higher priority channel can transfer data on the bus. Therefore, to prevent other channels from waiting for the bus for a long time, it is recommended to transmit data from memory to memory through channel 6 or channel 7.

3.5.4 Operating Mode

Initialization

To initialize the DMAC, do as follows:

Step 1 Configure DMAC_CONFIG to set the endianness of DMAC master 1 and DMAC master 2 and write 1 to DMAC_CONFIG[e] to enable the DMAC.

Step 2 Write 1 to all the bits of DMAC_INT_TC_CLR and DMAC_INT_ERR_CLR to clear all the interrupt status.

Step 3 Write 0 to the corresponding bits of DMAC_SYNC to set the DMA request signal groups to be synchronized.

Step 4 Configure and disable each channel in sequence. You can disable all the channels by writing 0 to DMAC_CX_CONFIG[e] of each channel.

----End

Enabling a Channel

After the DMAC is initialized, the DMAC can transmit data only when a DMAC channel is configured and enabled. To enable a DMA channel, do as follows:

Step 1 Read DMAC_ENBLD_CHNS to search for free channels and select one.

Step 2 Write 1 to the corresponding bits of DMAC_INT_TC_CLR and DMAC_INT_ERR_CLR to clear the interrupt status of the selected channel.

Step 3 Configure and enable the selected channel. Do as follows:

1. Configure DMAC_CX_SRC_ADDR to set the access start address of the source device.
2. Configure DMAC_CX_DEST_ADDR to set the access start address of the target device.
3. For single-block data transfer, write 0 to DMAC_CX_LLI.

4. For LLI data transfer, configure DMAC_CX_LLI as the LLI header pointer.
5. Write to DMAC_CX_CONTROL to set the master, data width, burst size, address increment, and transfer size of the source device and target device.
6. Write to DMAC_CX_CONFIG to set the DMA request signal, flow control mode, and interrupt mask of this channel. At this time, write 0 to DMAC_CX_CONFIG[e]. That is, this channel is not enabled currently.
7. Configure DMAC_CX_CONFIG to enable this channel. **Note:** In this case, change the channel enable bit to 1 but keep other bits unchanged.

----End

Processing an Interrupt

When an error occurs after or during the data transfer, the DMA channel reports an interrupt to the master ARM or slave ARM based on the channel configuration. To process an interrupt, do as follows:

Step 1 Read DMAC_INT_STAT to search for the channel that sends an interrupt request. If the interrupt is from ARM, read DMAC_INT_STAT1. When multiple channels send interrupt requests at the same time, the interrupt with the highest priority is processed first.

Step 2 Read DMAC_INT_TC_STAT (if the interrupt is from ARM, read DMAC_INT_STAT1) to check whether the selected bit is 1. If it is 1, it indicates that the interrupt sent from the corresponding channel is a transfer complete interrupt. If the value is 1, it indicates the interrupt is a terminal count interrupt. In this case go to Step 4; otherwise, go to Step 3.

Step 3 Read DMAC_INT_ERR_STAT (if the interrupt is from ARM, read DMAC_INT_STAT1) to check whether the selected bit is 1. The value 1 indicates that the interrupt sent from the corresponding channel is an error interrupt. If the selected bit is 1, it indicates the interrupt is an error interrupt. In this case, go to Step 5; otherwise, end the operation.

Step 4 Process the terminal count interrupt as follows:

1. Configure DMAC_INT_TC_CLR and write 1 to the selected bit to clear the interrupt status of the corresponding channel.
2. Take away or use up the data buffered in the memory. If required (such as for creating a buffer in the memory), configure and enable the channel again.
3. End the operation.

Step 5 Process the error interrupt as follows:

1. Configure DMAC_INT_ERR_CLR and write 1 to the selected bit to clear the interrupt status of the corresponding channel.
2. Check the error information. If required, configure and enable the channel again.
3. End the operation.

----End

3.5.5 Register Summary

Table 3-19 lists the DMAC registers.

Table 3-19 Summary of the DMAC registers (base address: 0x100D_0000)

Offset Address	Register	Description	Page
0x000	DMAC_INT_STAT	DMAC interrupt status register	3-40
0x004	DMAC_INT_TC_STAT	DMAC terminal count interrupt status register	3-41
0x008	DMAC_INT_TC_CLR	DMAC terminal count interrupt clear register	3-41
0x00C	DMAC_INT_ERR_STAT	DMAC error interrupt status register	3-42
0x010	DMAC_INT_ERR_CLR	DMAC error interrupt clear register	3-42
0x014	DMAC_RAW_INT_TC_STAT	DMAC raw terminal count interrupt status register	3-43
0x018	DMAC_RAW_INT_ERR_STAT	DMAC raw error interrupt status register	3-43
0x01C	DMAC_ENBLD_CHNS	DMAC channel enable status register	3-44
0x020	DMAC_SOFT_BREQ	DMA burst request register for software	3-44
0x024	DMAC_SOFT_SREQ	DMA single request register for software	3-45
0x028	DMAC_SOFT_LBREQ	DMA last burst request register for software	3-46
0x02C	DMAC_SOFT_LSREQ	DMA last single request register for software	3-47
0x030	DMAC_CONFIG	DMAC configuration register	3-47
0x034	DMAC_SYNC	DMAC request synchronization register	3-48
0x040	DMAC_INT_STAT1	DMAC interrupt status register	3-48
0x044	DMAC_INT_TC_STAT1	DMAC terminal count interrupt status register 1	3-49
0x048	DMAC_INT_ERROR_STAT1	DMAC error interrupt status register	3-49
0x100	DMAC_C0_SRC_ADDR	Source address register of channel 0	3-48
0x104	DMAC_C0_DEST_ADDR	Target address register of channel 0	3-51
0x108	DMAC_C0_LLI	LLI register of channel 0	3-51

Offset Address	Register	Description	Page
0x10C	DMAC_C0_CONTROL	Control register of channel 0	3-53
0x110	DMAC_C0_CONFIG	Configuration register of channel 0	3-56
0x120	DMAC_C1_SRC_ADDR	Source address register of channel 1	3-48
0x124	DMAC_C1_DEST_ADDR	Target address register of channel 1	3-51
0x128	DMAC_C1_LLI	LLI register of channel 1	3-51
0x12C	DMAC_C1_CONTROL	Control register of channel 1	3-53
0x130	DMAC_C1_CONFIG	Configuration register of channel 1	3-56
0x140	DMAC_C2_SRC_ADDR	Source address register of channel 2	3-48
0x144	DMAC_C2_DEST_ADDR	Target address register of channel 2	3-51
0x148	DMAC_C2_LLI	LLI register of channel 2	3-51
0x14C	DMAC_C2_CONTROL	Control register of channel 2	3-53
0x150	DMAC_C2_CONFIG	Configuration register of channel 2	3-56
0x160	DMAC_C3_SRC_ADDR	Source address register of channel 3	3-48
0x164	DMAC_C3_DEST_ADDR	Target address register of channel 3	3-51
0x168	DMAC_C3_LLI	LLI register of channel 3	3-51
0x16C	DMAC_C3_CONTROL	Control register of channel 3	3-53
0x170	DMAC_C3_CONFIG	Configuration register of channel 3	3-56
0x180	DMAC_C4_SRC_ADDR	Source address register of channel 4	3-48
0x184	DMAC_C4_DEST_ADDR	Target address register of channel 4	3-51
0x188	DMAC_C4_LLI	LLI register of channel 4	3-51
0x18C	DMAC_C4_CONTROL	Control register of channel 4	3-53
0x190	DMAC_C4_CONFIG	Configuration register of channel 4	3-56

Offset Address	Register	Description	Page
0x1A0	DMAC_C5_SRC_ADDR	Source address register of channel 5	3-48
0x1A4	DMAC_C5_DEST_ADDR	Target address register of channel 5	3-51
0x1A8	DMAC_C5_LLI	LLI register of channel 5	3-51
0x1AC	DMAC_C5_CONTROL	Control register of channel 5	3-53
0x1B0	DMAC_C5_CONFIG	Configuration register of channel 5	3-56
0x1C0	DMAC_C6_SRC_ADDR	Source address register of channel 6	3-48
0x1C4	DMAC_C6_DEST_ADDR	Target address register of channel 6	3-51
0x1C8	DMAC_C6_LLI	LLI register of channel 6	3-51
0x1CC	DMAC_C6_CONTROL	Control register of channel 6	3-53
0x1D0	DMAC_C6_CONFIG	Configuration register of channel 6	3-56
0x1E0	DMAC_C7_SRC_ADDR	Source address register of channel 7	3-48
0x1E4	DMAC_C7_DEST_ADDR	Target address register of channel 7	3-51
0x1E8	DMAC_C7_LLI	LLI register of channel 7	3-51
0x1EC	DMAC_C7_CONTROL	Control register of channel 7	3-53
0x1F0	DMAC_C7_CONFIG	Configuration register of channel 7	3-56

3.5.6 Register Description

DMAC_INT_STAT

DMAC_INT_STAT is the interrupt status register. It shows the master ARM the status of masked interrupts. If the certain bits of DMAC_INT_TC_STAT and DMAC_INT_ERR_STAT are masked at the same time, the corresponding bit of DMAC_INT_STAT is masked. Each bit of DMAC_INT_STAT maps to one DMAC channel.

Offset address	Register Name	Total Reset Value
0x000	DMAC_INT_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	int_stat	Status of the masked interrupts of each DMA channel. Bit[7:0] map to channels 7–0. 0: No interrupt is generated. 1: An interrupt is generated. The interrupt request may be an error interrupt or a terminal count interrupt.

DMAC_INT_TC_STAT

DMAC_INT_TC_STAT is the transfer complete interrupt status register. It shows the master ARM the status of masked transfer complete interrupts. The corresponding mask bit is DMAC_CX_CONFIG[itc]. X indicates the channel ID ranging from 0 to 7. This register must work with DMAC_INT_STAT.

Offset address	Register Name	Total Reset Value
0x004	DMAC_INT_TC_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	int_tc_stat	Status of the masked terminal count interrupts. Bit[7:0] map to channels 7–0. 0: No terminal count interrupt is generated. 1: A terminal count interrupt is generated.

DMAC_INT_TC_CLR

DMAC_INT_TC_CLR is the terminal count interrupt clear register. It is used to clear terminal count interrupts. Writing 1 to a certain bit clears both DMAC_INT_TC_STAT and DMAC_INT_TC_STAT1.

Offset address	Register Name	Total Reset Value
0x008	DMAC_INT_TC_CLR	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	WO	int_tc_clr	Terminal count interrupt clear. Bit[7:0] map to channels 7–0. 0: not cleared 1: cleared

DMAC_INT_ERR_STAT

DMAC_INT_ERR_STAT is the error interrupt status register. It shows the master ARM the status of masked error interrupts for querying. The corresponding mask bit is DMAC_CX_CONFIG[ie]. The register must work with DMAC_INT_STAT.

Offset address	Register Name	Total Reset Value
0x00C	DMAC_INT_ERR_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	int_err_stat	Status of masked error interrupts. Bit[7:0] map to channels 7–0. 0: No error interrupt is generated. 1: An error interrupt is generated.

DMAC_INT_ERR_CLR

DMAC_INT_ERR_CLR is an error interrupt clear register. It is used to clear error interrupts. Writing 1 to a certain bit clears both DMAC_INT_ERR_STAT and DMAC_INT_ERR_STAT1.

Offset address	Register Name	Total Reset Value
0x010	DMAC_INT_ERR_CLR	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	WO	int_err_clr	Error interrupt clear. Bit[7:0] map to channels 7–0. 0: not cleared 1: cleared

DMAC_RAW_INT_TC_STAT

DMAC_RAW_INT_TC_STAT is the raw terminal count interrupt status register. It shows the status of the raw terminal count interrupt of each channel.

Offset address	Register Name	Total Reset Value
0x014	DMAC_RAW_INT_TC_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	raw_int_tc_stat	Status of the raw terminal count interrupt of each channel. Bit[7:0] map to channels 7–0. 0: No terminal count interrupt is generated. 1: A terminal count interrupt is generated.

DMAC_RAW_INT_ERR_STAT

DMAC_RAW_INT_ERR_STAT is the raw error interrupt status register. It shows the status of the raw error interrupt of each channel.

Offset address	Register Name	Total Reset Value
0x018	DMAC_RAW_INT_ERR_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	raw_int_err_stat	Status of the raw error interrupt of each channel. Bit[7:0] map to channels 7–0. 0: No error interrupt is generated. 1: An error interrupt is generated.

DMAC_ENBLD_CHNS

DMAC_ENBLD_CHNS is the channel enable register that shows the enabled channels.

For example, if a certain bit of DMAC_ENBLD_CHNS is 1, it indicates that the corresponding channel is enabled. The enable bit of DMAC_CX_CONFIG of each channel determines whether the corresponding channel is enabled. When the DMA transfer of a channel is complete, the bit of DMAC_ENBLD_CHNS corresponding to the channel is cleared.

Offset address	Register Name	Total Reset Value
0x01C	DMAC_ENBLD_CHNS	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	enabled_channels	Channel enable. Bit[7:0] map to channels 7–0. 0: disabled 1: enabled

DMAC_SOFT_BREQ

DMAC_SOFT_BREQ is the burst request register for software. It is used to control whether to generate DMA burst requests through the software.

When this register is read, the device that is requesting the DMA burst transfer can be queried. This register and any peripheral each can generate a DAM request.

NOTE
Do not use a software DMA request and a hardware DMA request at the same time.

Offset address	Register Name	Total Reset Value
0x020	DMAC_SOFT_BREQ	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	RW	soft_breq	Controls whether to generate DMA burst transfer requests through the software. For the request signal corresponding to each bit, see Table 3-18. When the register is written: 0: no effect 1: A DMA burst transfer request is generated. When the transfer is complete, the corresponding bit is cleared. When the register is read: 0: The peripheral corresponding to the request signal DMACBREQ[15:0] does not send a DMA burst request. 1: The peripheral corresponding to the request signal DMACBREQ[15:0] is requesting the DMA burst transfer.

DMAC_SOFT_SREQ

DMAC_SOFT_SREQ is the DMA single request register for software. It is used to control whether to generate a DMA single transfer request through the software.

When this register is read, the device that is requesting the DMA single transfer can be queried. This register and the 16 DMA request input signals of the DMAC each can generate a DMA request.

NOTE
Do not use a software DMA request and a hardware DMA request at the same time.

Offset address	Register Name	Total Reset Value
0x024	DMAC_SOFT_SREQ	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	RW	soft_sreq	Controls whether to generate DMA single transfer requests through the software. For the request signal corresponding to each bit, see Table 3-18. When the register is written: 0: no effect 1: A DMA single transfer request is generated. When the transfer is complete, the corresponding bit is cleared. When the register is read: 0: The peripheral corresponding to the request signal DMACBREQ[15:0] does not send a DMA signal request. 1: The peripheral corresponding to the request signal DMACBREQ[15:0] is requesting the single DMA transfer.

DMAC_SOFT_LBREQ

DMAC_SOFT_LBREQ is the last burst request register for software. It is used to control whether to generate a DMA last burst transfer request through the software.

Offset address	Register Name	Total Reset Value
0x028	DMAC_SOFT_LBREQ	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	WO	soft_lbreq	Controls whether to generate a last burst request through the software. For details about the request signal corresponding to each bit, see Table 3-18. 0: no effect 1: A DMA last burst transfer request is generated. When the transfer is complete, the corresponding bit is cleared.

DMAC_SOFT_LSREQ

DMAC_SOFT_LSREQ is the DMA last single request register for software. It is used to control whether to generate the last DMA single transfer request through the software.

Offset address	Register Name	Total Reset Value
0x02C	DMAC_SOFT_LSREQ	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	WO	soft_lsreq	Controls whether to generate the last single transfer request through the software. For details about the request signal corresponding to each bit, see Table 3-18. 0: no effect 1: The last DMA single transfer request is generated. When the transfer is complete, the corresponding bit is cleared.

DMAC_CONFIG

DMAC_CONFIG is the configuration register. It is used to configure the DMAC. You can change the endianness mode of the two master interfaces of the DMAC by writing m1 (bit[1]) and m2 (bit[2]) of this register. After reset, the two master interfaces are set to little endian mode.

NOTE

The two master interfaces work in little endian mode.

Offset address	Register Name	Total Reset Value
0x030	DMAC_CONFIG	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																													m2	m1	e
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:3]	RO	reserved	Reserved.
[2]	RW	m2	Endianness mode of master 2. 0: little endian mode 1: big endian mode

[1]	RW	m1	Endianness mode of master 1. 0: little endian mode 1: big endian mode
[0]	RW	e	DMAC enable. 0: disabled 1: enabled

DMAC_SYNC

DMAC_SYNC is the synchronization register. It is used to enable or disable the synchronization logics provided for DMA request signals.

NOTE

It is recommended to disable the synchronization logics provided for all the request signals.

Offset address	Register Name	Total Reset Value
0x034	DMAC_SYNC	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	RW	dmac_sync	Controls whether to synchronize request signals. For the request signal corresponding to each bit, see Table 3-18 0: Enable the synchronization logics provided for the DMA request signals of the corresponding peripheral. 1: Disable the synchronization logics provided for the DMA request signals of the corresponding peripheral

DMAC_INT_STAT1

DMAC_INT_STAT1 is the interrupt status register. It shows the slave ARM the status of masked interrupts. If the certain bits of DMAC_INT_STAT1 and DMAC_INT_ERR_STAT1 are masked at the same time, the corresponding bit of DMAC_INT_STAT1 is masked. Each bit of DMAC_INT_STAT1 maps to one DMAC channel.

Offset address	Register Name	Total Reset Value
0x040	DMAC_INT_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	int_stat	Status of the masked interrupts of each DMA channel. Bit[7:0] map to channels 7–0. 0: No interrupt is generated. 1: An interrupt is generated. The interrupt request may be an error interrupt or a terminal count interrupt.

DMAC_INT_TC_STAT1

DMAC_INT_TC_STAT1 is the terminal count interrupt status register. It shows the slave ARM the status of masked terminal count interrupts for querying. The corresponding mask bit is DMAC_CX_CONFIG[itc]. X indicates the channel ID ranging from 0 to 7. This register must work with DMAC_INT_STAT1.

Offset address	Register Name	Total Reset Value
0x044	DMAC_INT_TC_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	int_tc_stat	Status of the masked terminal count interrupts. Bit[7:0] map to channels 7–0. 0: No terminal count interrupt is generated. 1: A terminal count interrupt is generated.

DMAC_INT_ERR_STAT1

DMAC_INT_ERR_STAT1 is the error interrupt status register. It shows the slave ARM the status of masked error interrupts for querying. The corresponding mask bit is DMAC_CX_CONFIG[ie]. The register must work with DMAC_INT_STAT1.

Offset address	Register Name	Total Reset Value
0x048	DMAC_INT_ERR_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7:0]	RO	int_err_stat	Status of masked error interrupts. Bit[7:0] map to channels 7–0. 0: No error interrupt is generated. 1: An error interrupt is generated.

DMAC_CX_SRC_ADDR

DMAC_CX_SRC_ADDR is the source address register. It shows the source addresses (sorted by byte) of the data to be transmitted.

Its offset address is 0x100+Xx0x20. The value of X ranges from 0 to 7. The values 0–7 map to DMA channels 0–7.

Before a channel is enabled, its register must be programmed through software. After the channel is enabled, the register is updated in the following cases:

- When the source address is incremented.
- When a complete data block is transferred and then loaded from LLI nodes.

When a channel is active, no valid information is obtained when this register is read. This is because that after software obtains the register value, the value is changed during data transfer. Therefore, the register is read after the channel stops data transfer. At this time, the read value is the last source address read by the DMAC.

The source and target addresses must be aligned with the transfer widths of the source and target devices.

Offset address	Register Name	Total Reset Value
0x100+X x 0x20	DMAC_CX_SRC_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	src_addr	DMA source address.

NOTE

- The DMAC provides eight channels. Each channel has the following five channel registers:
- DMAC_CX_SRC_ADDR
- DMAC_CX_DEST_ADDR
- DMAC_CX_LLI
- DMAC_CX_CONTROL
- DMAC_CX_CONFIG

When the DMA loads LLI nodes from a memory, the DMAC updates DMAC_CX_SRC_ADDR the preceding five registers excluding DMAC_CX_CONFIG automatically.

CAUTION

During DMA transfer, the DMAC may perform unpredictable operations when the preceding channel registers are updated. Before changing the settings of a channel, you must disable the channel and then configure its related registers.

DMAC_CX_DEST_ADDR

DMAC_CX_DEST_ADDR is the target address register. Its offset address is 0x104+Xx0x20. The value of X ranges from 0 to 7. The values 0–7 map to DMA channels 0–7.

This register contains the target address (sorted by byte) of the data to be transmitted. Before a channel is enabled, its register must be programmed through software. After the channel is enabled, the register is updated in the following cases:

- Target address increment.
- When a complete data block is transferred and then loaded from LLI nodes.

When a channel is active, no valid information is obtained when this register is read. This is because that after software obtains the register value, the value is changed during data transfer. Therefore, the register is read after the channel stops data transfer. At this time, the read value is the last target address written by the DMAC.

Offset address	Register Name	Total Reset Value
0x104+X x 0x20	DMAC_CX_DEST_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	dest_addr	DMA target address.

DMAC_CX_LLI

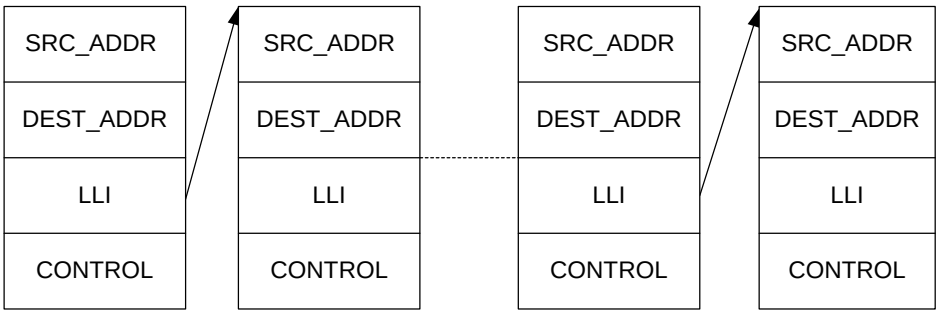
DMAC_CXLLI is the LLI register. Its offset address is 0x108+Xx0x20. The value of X ranges from 0 to 7. The values 0–7 map to DMA channels 0–7.

The data structure of the DMAC LLI node is as follows:

- Channel register DMAC_CX_SRC_ADDR, for setting the start address of the source device.
- Channel register DMAC_CX_DEST_ADDR, for setting the start address of the target device.
- Channel register DMAC_CX_LLI, for setting the address of the next node.
- Channel register DMAC_CX_CONTROL, for setting the master, data width, burst size, address increment, and transfer size of the source device and target device.

Figure 3-10 shows the structure of the DMAC LLIs.

Figure 3-10 Structure of the DMAC LLIs



CAUTION

The value of the LLI field must be equal to or less than 0xFFFF_FFF0. Otherwise, the address is rolled back to 0x0000_0000 during a 4-word burst transfer. As a result, the data structure of LLI nodes cannot be stored in a continuous address area.

If the LLI field is set to 0, it indicates that the current node is at the end of the linked list. In this case, the channel is disabled after the corresponding data blocks of the current node are transferred.

Offset address	Register Name	Total Reset Value
0x108+Xx0x20	DMAC_CXLLI	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	RW	lli	LLI. The [31:2] bit of the next LLI node address and the address bit[1:0] are both set to 0. A linked list address must be 4-byte aligned.

[1]	RO	reserved	Reserved. This bit must be 0 during writes and must be masked during reads.
[0]	RW	lm	Master for loading the next LLI node. 0: Master1 1: Master2

DMAC_CX_CONTROL

DMAC_CX_CONTROL is the channel control register. Its offset address is 0x10C+Xx0x20. The value of X ranges from 0 to 7. The values 0–7 map to DMA channels 0–7.

This register contains the control information about the DMA channels, such as the transfer size, burst size, and transfer bit width.

Before a channel is enabled, its register must be programmed through software. When the channel is enabled, the value of the register is updated when being loaded from the LLI node after a complete data block is transferred.

When a channel is active, no valid information is obtained when this register is read. because after software obtains the register value, the value is changed during data transfer. After the channel stops data transfer, the register can be read.

Offset address	Register Name	Total Reset Value
0x10C+Xx 0x20	DMAC_CX_CONTROL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i	prot			di	si	d	s	dwidth			swidth			dbsize			sbsize			transfersize												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	i	Terminal count interrupt enable. This bit determines whether the current LLI node triggers a terminal count interrupt. 0: do not trigger 1: trigger
[30:28]	RW	prot	Access protection HPROT[2:0] signal sent by a master. For details about these bits, see Table 3-22.
[27]	RW	di	Target address increment. 0: The target address is not incremented 1: The target address is incremented once after a data segment is transferred If the target device is a peripheral, the target address is not incremented. If the target device is a memory, the target address is incremented.
[26]	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.
[25]	RW	d	Master for accessing the target device. 0: Master 1 accesses SIO0, SIO1, UART0, SDIO, and SPI. 1: Master 2 accesses the NOR flash and DDR.
[24]	RW	s	Master for accessing the source device. 0: Master 1 accesses SIO0, SIO1, UART0, SDIO, and SPI. 1: Master 2 accesses the NOR flash and DDR.
[23:21]	RW	dwidth	Transfer bit width of the target device. The transfer bit width is invalid if it is greater than the bit width of the master. The data widths of the target and source devices can be different. The hardware automatically packs and unpacks the data. For the mapping between the value of DWidth and the bit width, see Table 3-21.
[20:18]	RW	swidth	Transfer bit width of the source device. The transfer bit width is invalid if it is greater than the bit width of the master. The data widths of the target and source devices can be different. The hardware automatically packs and unpacks the data. For the mapping between the value of SWidth and the bit width, see Table 3-21.
[17:15]	RW	dbsize	Burst size of the target device. It indicates the number of data segments to be transferred by the target device in a burst transfer, that is, the number of transferred data segments when DMACCxBREQ is valid. This value must be set to a burst size supported by the target device. If the target device is a memory, the value is set to the storage address boundary. For the mapping between the value of DBSize and the transfer size, see Table 3-20.
[14:12]	RW	sbsize	Burst size of the source device. It indicates the number of data segments to be transferred by the source device in a burst transfer, that is, the number of transferred data segments when DMACCxBREQ is valid. The value must be set to a burst size supported by the source device. If the source device is a memory, the value is set to the storage address boundary. For the mapping between the value of SBSize and the transfer size, see Table 3-20.
[11:0]	RW	transfersize	If the DMAC is a flow controller, the DMA transfer size can be

			configured by writing DMAC_CX_CONTROL. The transfer size indicates the number of the data segments to be transferred by the source device. When DMAC_CX_CONTROL is read, the number of data segments transferred through the bus connected to the target device is obtained. When a channel is active, no valid information is obtained when this register is read. This is because that after software obtains the register value, the value is changed during data transfer. Therefore, the register is read after the channel is enabled and data transfer is stopped.

Table 3-20 lists the value of DBSize or SBSize of DMAC_CX_CONTROL and the corresponding burst size.

Table 3-20 Mapping between the value of DBSize or SBSize and the burst size

DBSize or SBSize	Burst Size
000	1
001	4
010	8
011	16
100	32
101	64
110	128
111	256

Table 3-21 lists the value of DWidth or SWidth of DMAC_CX_CONTROL and the corresponding transfer data width.

Table 3-21 Mapping between the value of DWidth or SWidth and the transfer bit width

SWidth or DWidth	Transfer Bit Width
000	Byte (8 bits)
001	Halfword (16 bits)
010	Word (32 bits)
Others	Reserved.

Note the following points when configuring DMAC_CX_CONTROL register:

- When the transfer bit width of the source device is less than that of the target device, the product of the transfer bit width and transfer size of the source device must be an integer multiple of the transfer bit width of the target device. Otherwise, the data in the FIFO is stranded and then lost.
- SWidth and DWidth fields cannot be set to undefined bit widths.
- If the transfer size field is set to 0 and the DMAC is a flow controller, the DMAC does not transfer data. In this case, the programmer needs to disable the DMA channel and reprogram it.
- Do not perform common write/read tests on the DMAC_CX_CONTROL register, because the transfer size field is not a common register field in which the written value is the same as the read value. During writes, this field serves as a control register, because it determines the number of data segments transferred by the DMAC. During reads, this field serves as a status register, because it returns the number (in the unit of the bit width of the source device) of the remaining data segments to be transferred.
- When the transfer size field is set to a value greater than the depth of the FIFO (peripheral FIFO, not DMAC FIFO) of the source device or target device, the source address or target address of the DMAC must be set to the non-increment mode. Otherwise, the peripheral FIFO may overflow.

The bus access information is provided to the source device or target device by the master interface signals during data transfer. Such information is related to the bits DMAC_CX_CONTROL[prot] and DMAC_CX_CONFIG[Lock] that are set by programming the channel registers. Table 3-22 describes the three protection bits of the prot field of DMAC_CX_CONTROL.

Table 3-22 Attributes and definitions of the prot field of DMAC_CX_CONTROL

Bit	Description	Function
[2]	Cacheable or nonCacheable	Indicates whether the access is cacheable. 0: noncacheable 1: cacheable This bit controls the output of the bus signal HPROT[3].
[1]	bufferable or nonbufferable	Indicates whether the access is bufferable. 0: nonbufferable 1: bufferable This bit controls the output of the bus signal HPROT[2].
[0]	privileged or user	Access mode. 0: user mode 1: privileged mode This bit controls the output of the bus signal HPROT[1].

DMAC_CX_CONFIG

DMAC_CX_CONFIG is the channel configuration register. Its offset address is 0x110+Xx0x20. The value of X ranges from 0 to 7. The values 0–7 map to DMA channels 0–7.

This register is not updated when a new LLI node is loaded.

Offset address	Register Name	Total Reset Value
0x110+Xx0x20	DMAC_CX_CONFIG	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved													h	a	l	itc	ie	flow_cntrl			reserved	dest_peripheral				reserved	src_peripheral			e
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	RO	reserved	Reserved. This bit must be 0 during writes and must be masked during reads.
[20]	ITC1	R/W	Terminal count interrupt mask bit. 0: Mask the terminal count interrupt of the channel of the slave ARM. 1: Do not mask the terminal count interrupt of the channel of the slave ARM.
[19]	IE1	R/W	Error interrupt mask bit. 0: Mask the error interrupt of the channel of the slave ARM. 1: Do not mask the error interrupt of the channel of the slave ARM.
[17]	RO	a	Active bit. 0: There is no data in the channel FIFO. 1: There is data in the channel FIFO. This bit can disable a DMA channel without data loss by working with the halt bit and the channel enable bit.
[16]	RW	l	Lock bit. 0: Disable lock transfer on the bus. 1: Enable lock transfer on the bus.
[15]	RW	itc	Terminal count interrupt mask bit. 0: Mask the terminal count interrupt of the channel of the master ARM. 1: Do not mask the terminal count interrupt of the channel of the master ARM.
[14]	RW	ie	Error interrupt mask bit. 0: Mask the error interrupt of the channel of the master ARM. 1: Do not mask the error interrupt of the channel of the master ARM.
[13:11]	RW	flow_cntrl	Flow control and transfer type field. This field is used to specify the flow controller and the transfer

			type. The flow controller can be the DMAC, source device, or target device. The transfer type can be memory to peripheral, peripheral to memory, peripheral to peripheral, or memory to memory. For details, see Table 3-23.
[10]	RO	reserved	Reserved. This bit must be 0 during writes and must be masked during reads.
[9:6]	RW	dest_peripheral	Target device. This field is used to select a peripheral request signal as the request signal of the DMA target device of the channel. If the target device for DMA transfer is a memory, this field is ignored.
[5]	RO	reserved	Reserved. This bit must be 0 in writes and must be masked in reads.
[4:1]	RW	src_peripheral	Source device. This field is used to select a peripheral request signal as the request signal of the DMA source device of the channel. If the source device for DMA transfer is a memory, this field is ignored.
[0]	RW	e	Channel enable bit. The channel status can be queried by reading DMAC_CX_CONFIG or DMACEnbldChns. 0: disabled 1: enabled Clearing this bit can disable a channel. When this bit is cleared, the current bus transfer continues until all data is transferred. Then, the channel is disabled and the remaining data in the FIFO is lost. When the last LLI is transferred or an error occurs during transfer, the channel is also disabled and this bit is cleared. If you want to disable a channel without data loss, the halt bit must be set to 1, so the subsequent DMA requests are ignored by the channel. After this, the active bit must be polled until its value becomes 0. The value 0 indicates that there is no data in the channel FIFO. In this case, the enable bit can be cleared. In case of enabling a channel by setting this bit to 1, the channel can be enabled only after it is initialized again. If a channel is enabled by setting this bit to 1 only, unpredictable results may occur.

When a channel is disabled by writing to the channel enable bit, this bit can be set to 1 again only after the corresponding bit of DMAC_ENBLD_CHNS is polled to be 0. This is because the channel is not disabled immediately after the channel enable bit is cleared. In addition, the running delay during a bus burst operation should be considered.

Table 3-23 lists the flow controller and transfer types corresponding to the flow_cntrl field of DMAC_CX_CONFIG.

Table 3-23 Flow controller and transfer types corresponding to the flow_cntrl field

Bit Value	Transfer Type	Controller
000	Memory to memory	DMAC
001	Memory to peripheral	DMAC
010	Peripheral to memory	DMAC
011	Source device to target device	DMAC
Others	Reserved.	Reserved.

3.6 CIPHER

3.6.1 Overview

The CIPHER module is an encryption and decryption module that uses the data encryption standard (DES), 3DES, and advanced encryption standard (AES) algorithms. The DES or 3DES algorithm is based on the FIPS46-3 standard, whereas the AES algorithm is based on the FIPS 197 standard. The operating modes of the DES or 3DES algorithm comply with the FIPS-81 standard and the operating modes of the AES algorithm comply with the NIST special800-38a standard.

The CIPHER module can encrypt or decrypt a large amount of data effectively. In addition, it encrypts and decrypts one or multiple blocks at a time.

3.6.2 Features

The CIPHER module has the following features:

- Supports the AES key length of 128 bits, 192 bit, or 256 bits.
- Supports the DES key length of 64 bits.
- Supports 3-key and 2-key modes for the 3DES algorithm.
- Supports the operating modes of electronic codebook (ECB), cipher block chaining (CBC), 1-/8-/128-cipher feedback (CFB), 128-output feedback (OFB), and counter (CTR) for the AES algorithm. These operating modes comply with the NIST special800-38a standard.
- Supports the operating modes of ECB, CBC, 1/8/64-CFB, and 1/8/64-OFB for the DES or 3DES algorithm. These operating modes comply with the FIPS-81 standard.
- Encrypts and decrypts one or multiple blocks at a time in ECB, CBC, CFB, or OFB operating mode.
- Encrypts and decrypts a block at a time in CTR operating mode of the AES algorithm.
- Supports the functions of interrupt status query, interrupt mask, and interrupt clear.
- Supports the function of forcible abort.
- Adjusts the byte sequences of the input data (including the block input, vector input, and key) and the output data (including the block output and vector output).

3.6.3 Function Description

The operating modes of the DES or 3DES algorithm comply with the FIPS-81 standard and the operating modes of the AES algorithm comply with the NIST special800-38a standard. For the DES, 3DES, and AES algorithms, the ECB, CBC, and CFB operating modes are the same, whereas the CTR and OFB operating modes are different. Note that the CTR operating mode is exclusive for the AES algorithm only.

3DES Algorithm

The 3DES algorithm supports both 3-key and 2-key operations. The 2-key operation can be regarded as a simplified 3-key operation. To be specific, for the 2-key operation, key 3 is replaced with key 1.

Figure 3-11 shows the 3DES encryption of the 3-key operation and the 2-key operation.

Figure 3-11 3DES encryption of the 3-key operation and the 2-key operation

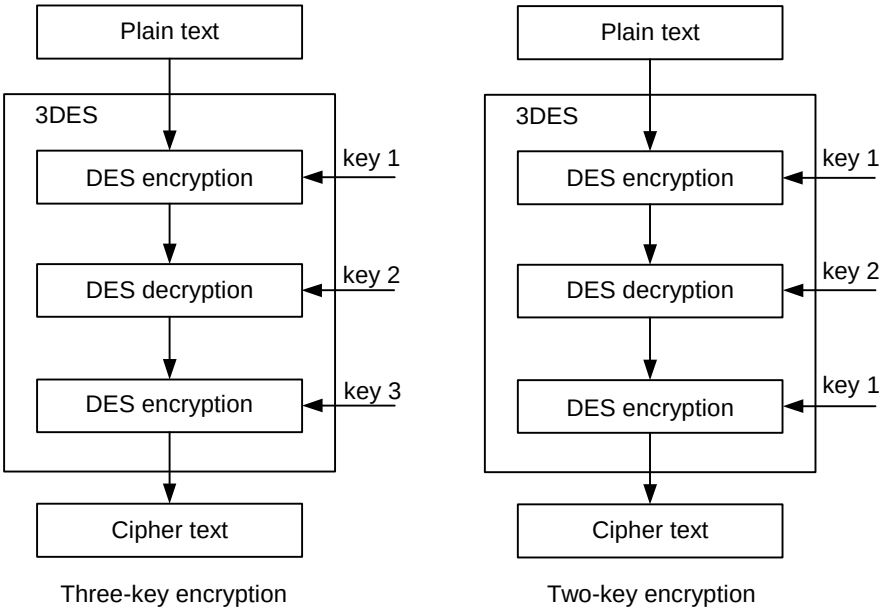
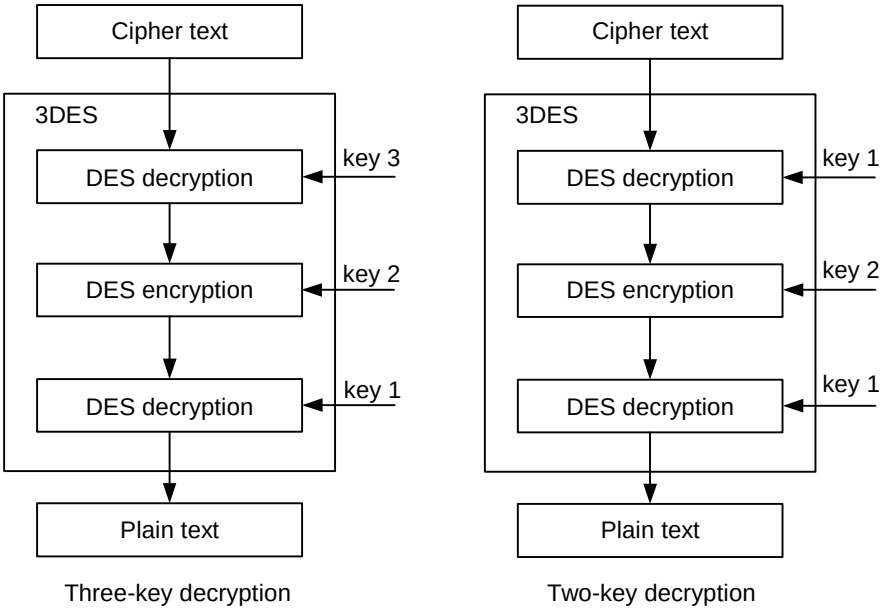


Figure 3-12 shows the 3DES decryption of the 3-key operation and the 2-key operation.

Figure 3-12 3DES decryption of the 3-key operation and 2-key operation



ECB Mode

In ECB mode, the encryption and decryption algorithms are directly applied to the block data. In addition, the operation of each block is independent. With this feature, the plain text encryption and cipher text decryption can be performed concurrently. Figure 3-13 shows the ECB mode of the AES and DES algorithms and Figure 3-14 shows the ECB mode of the 3DES algorithm.

Figure 3-13 ECB mode of the AES and DES algorithms

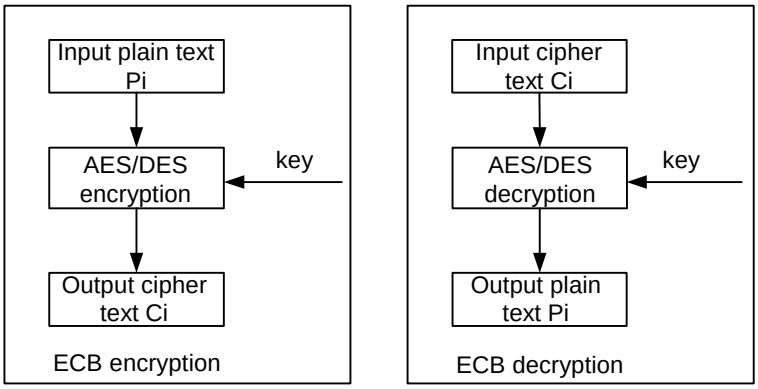
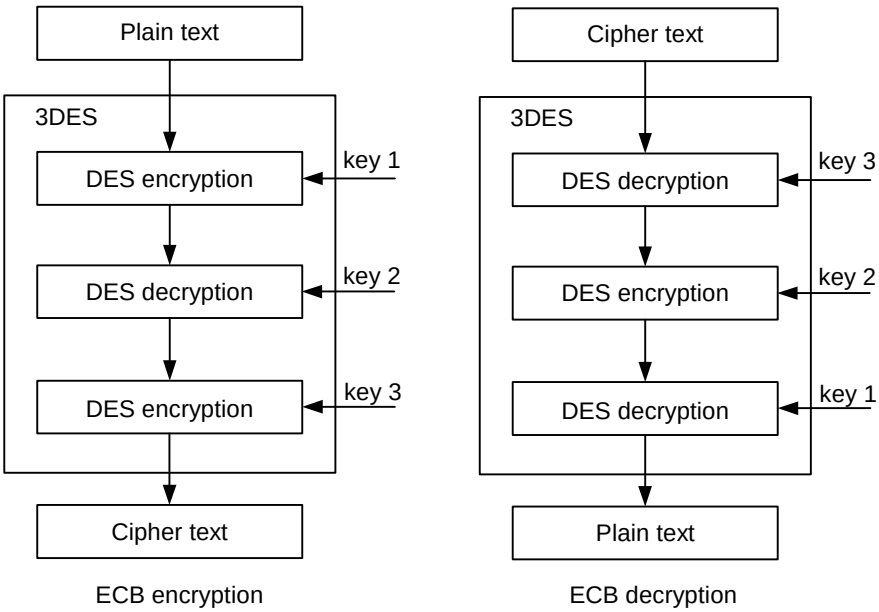


Figure 3-14 ECB mode of the 3DES algorithm



CBC Mode

In CBC mode, the encryption operation is performed only after the encrypted input plain text block is exclusive-ORed with the input vector (IV). In addition, each plain text block is encrypted based on the processing result (namely, cipher text) of the previous plain text block. Therefore, encryption operations cannot be performed concurrently in CBC mode. The decryption operation, however, is independent of output plain text of the previous block. Therefore, decryption operations can be performed concurrently. Figure 3-15 shows the CBC mode of the AES and DES algorithms and Figure 3-16 shows the CBC mode of the 3DES algorithm.

Figure 3-15 CBC mode of the AES and DES algorithms

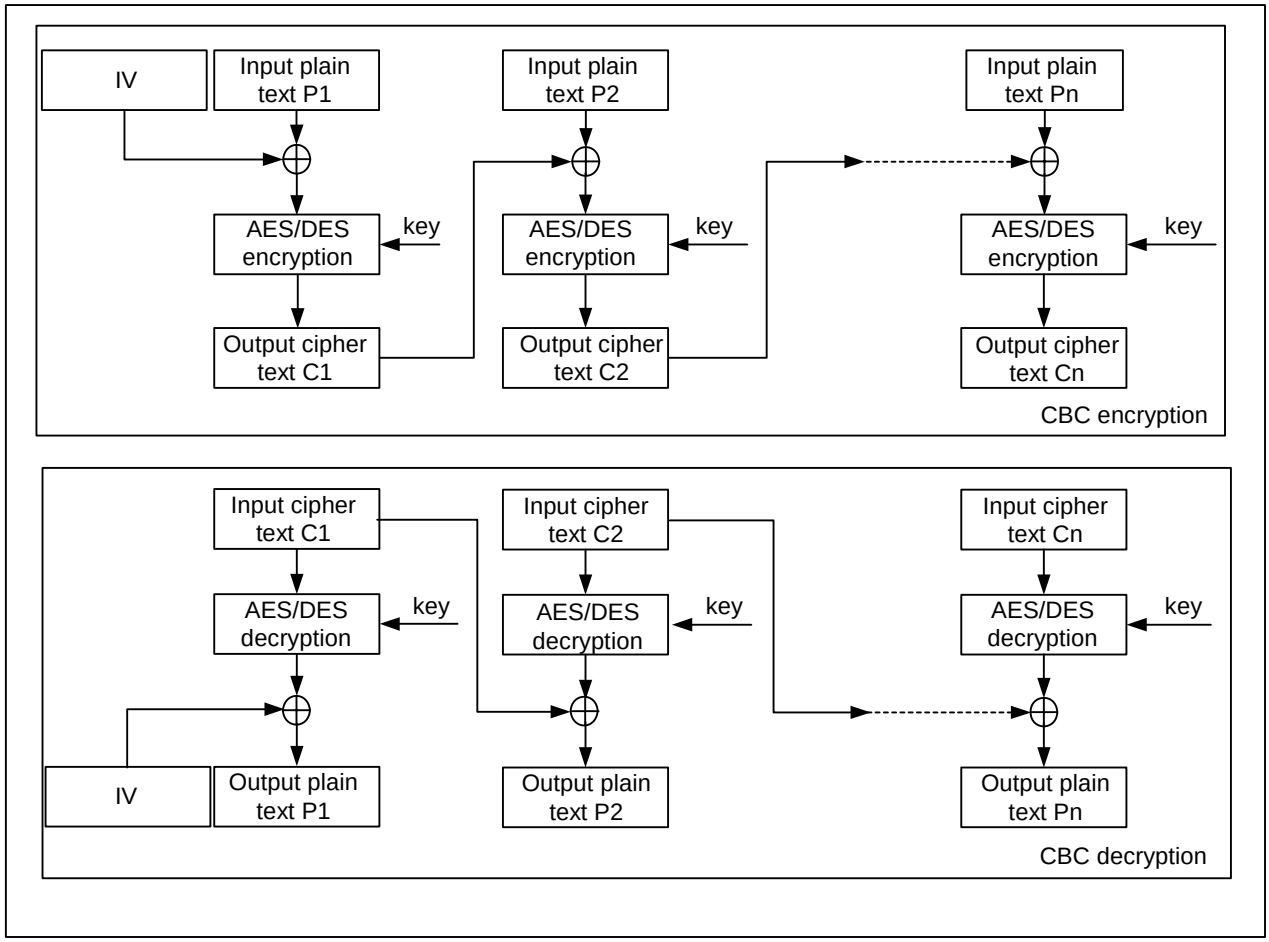
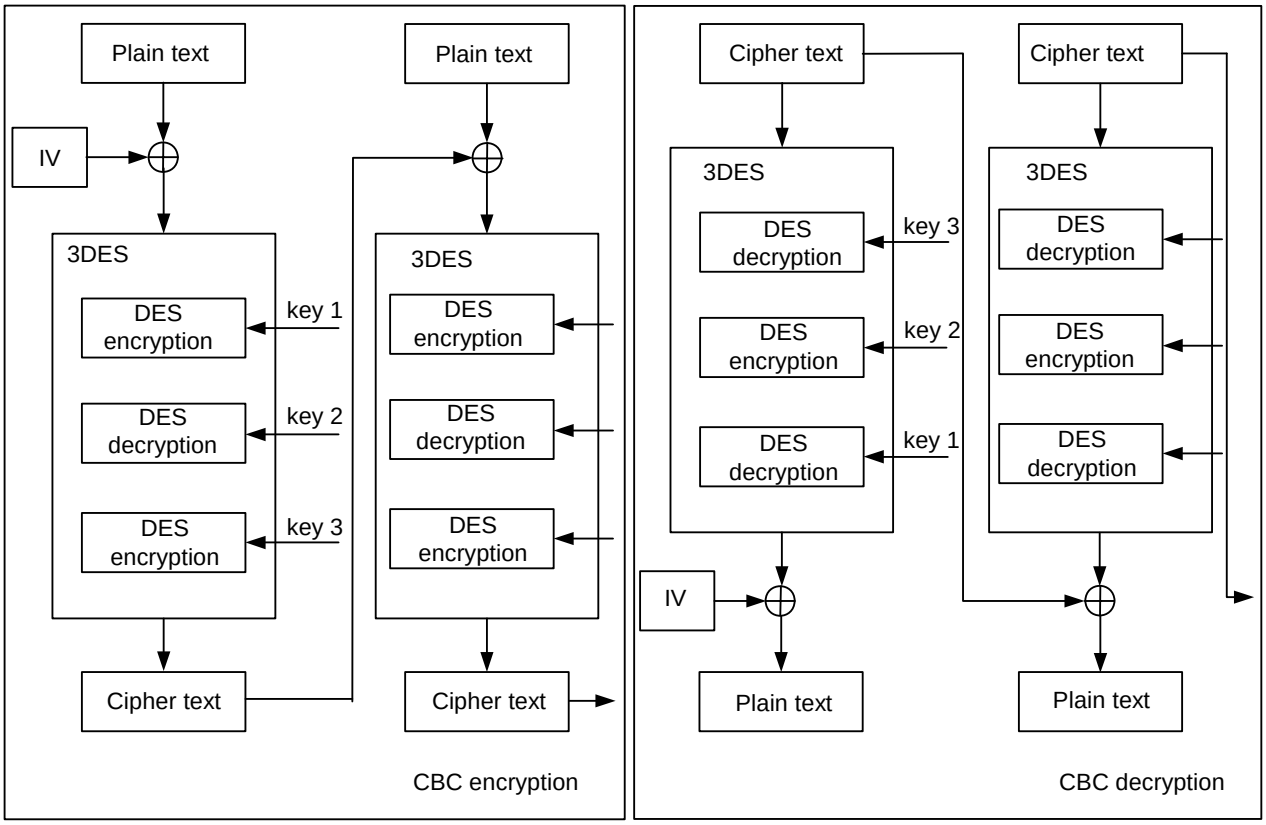


Figure 3-16 CBC mode of the 3DES algorithm



CFB Mode

The CFB mode is used to convert a block cipher into a stream cipher. This mode is implemented by selecting the operation bits of the CFB. The shift operation bits are represented by *s* in either of the following cases:

- For the DES or 3DES algorithm, s bits can be 1 bit, 8 bits, or 64 bits.
- For the AES algorithm, s bits can be 1 bit, 8 bits, or 128 bits.

Figure 3-17 shows the s-bit CFB mode of the AES and DES algorithm and Figure 3-18 shows the s-bit CFB mode of the 3DES algorithm.

Figure 3-17 S-bit CFB mode of the AES and DES algorithms

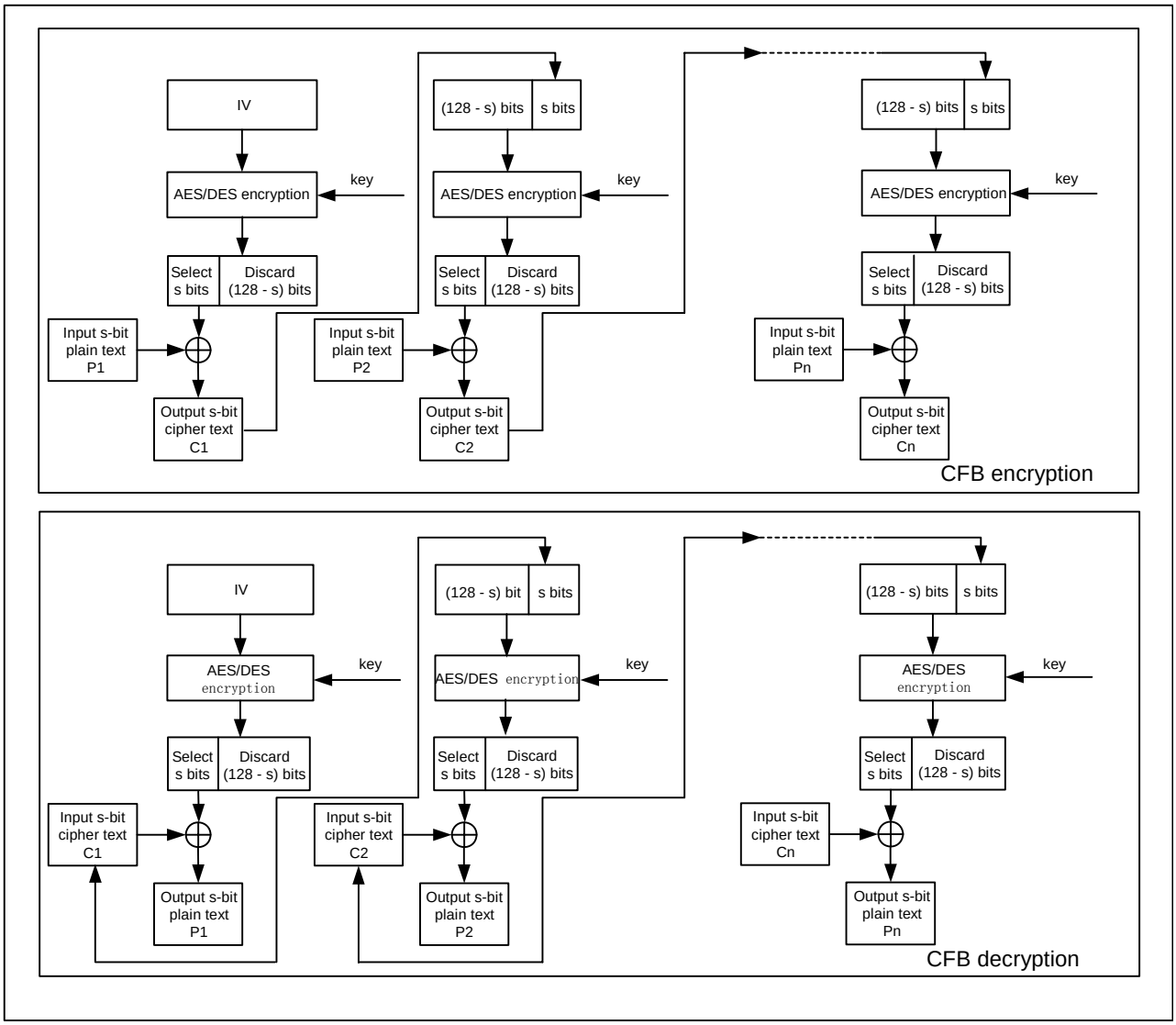
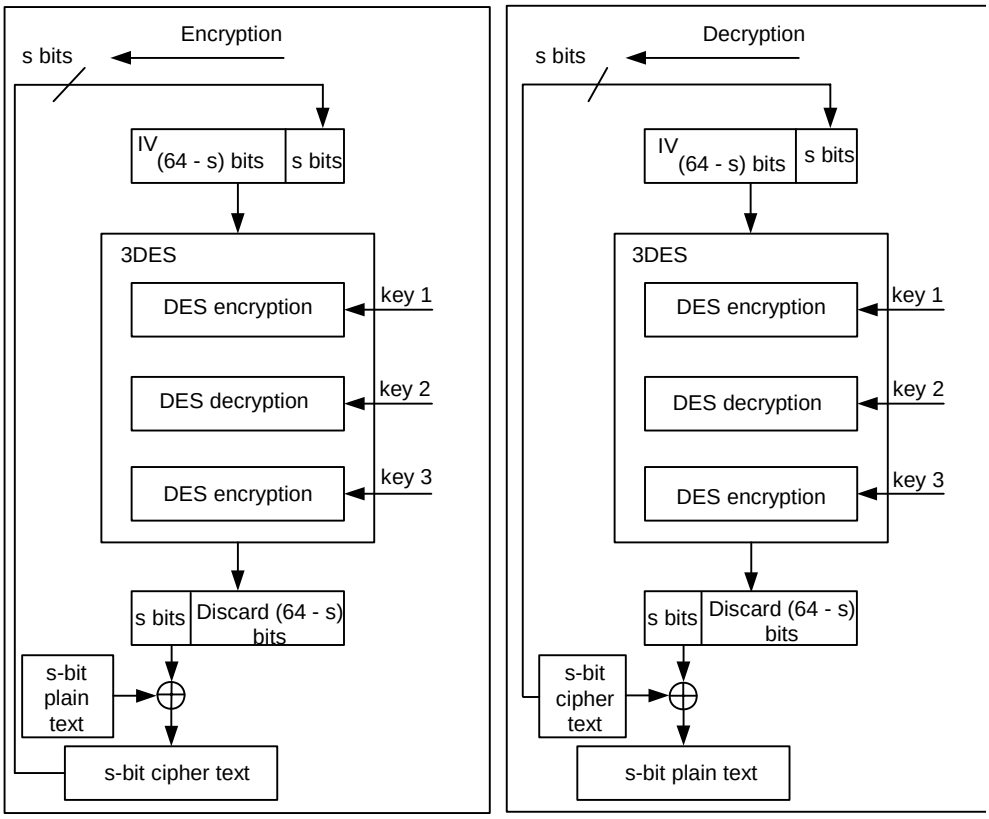


Figure 3-18 S-bit CFB mode of the 3DES algorithm



OFB Mode

In OFB mode, IVs serve as the inputs during encryption. If a same key is used repeatedly, different IVs must be used to ensure operation security. The s bits can be the following values:

- For the DES or 3DES algorithm, s bits can be 1 bit, 8 bits, or 64 bits.
- For the AES algorithm, s bits must be 128 bits.

Figure 3-19 shows the OFB mode of the AES algorithm.

Figure 3-19 OFB mode of the AES algorithm

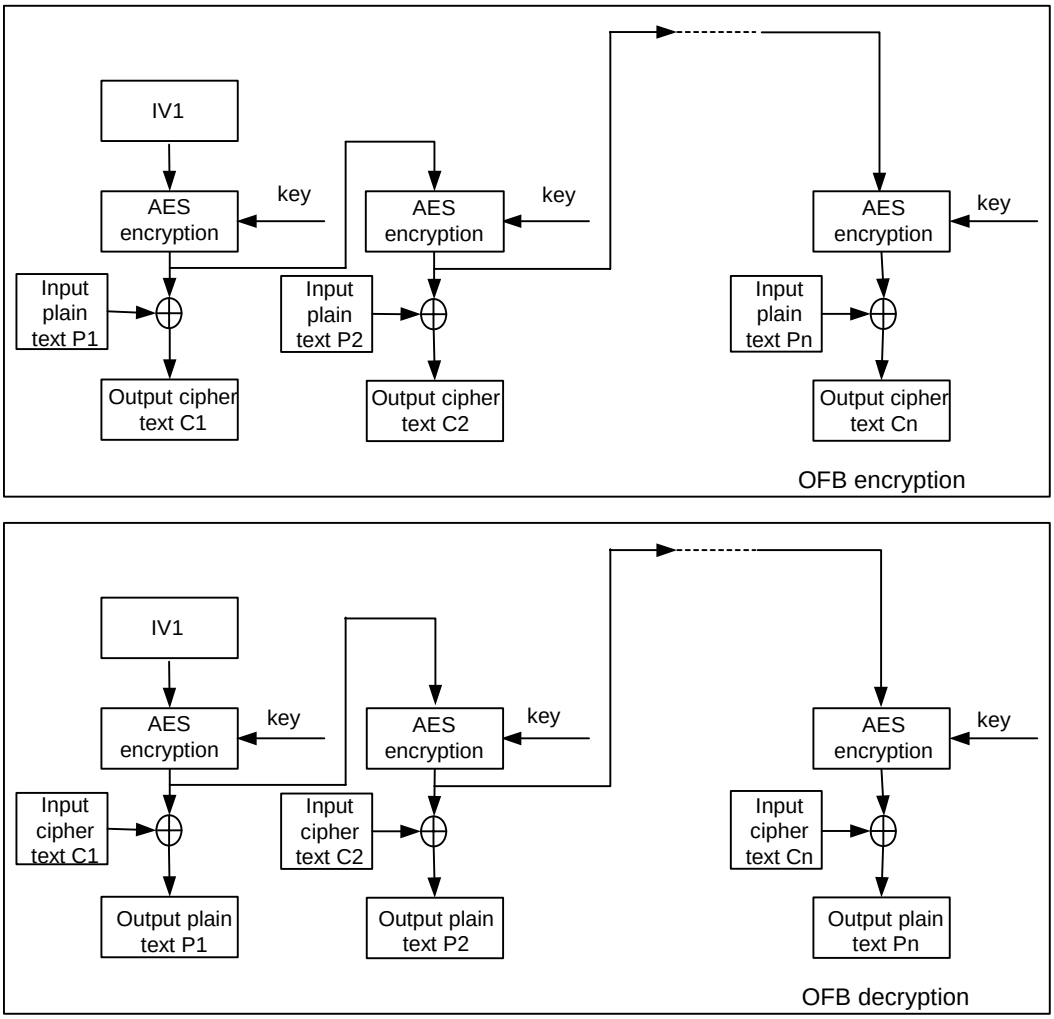


Figure 3-20 shows the s-bit OFB mode of the DES algorithm.

Figure 3-20 S-bit OFB mode of the DES algorithm

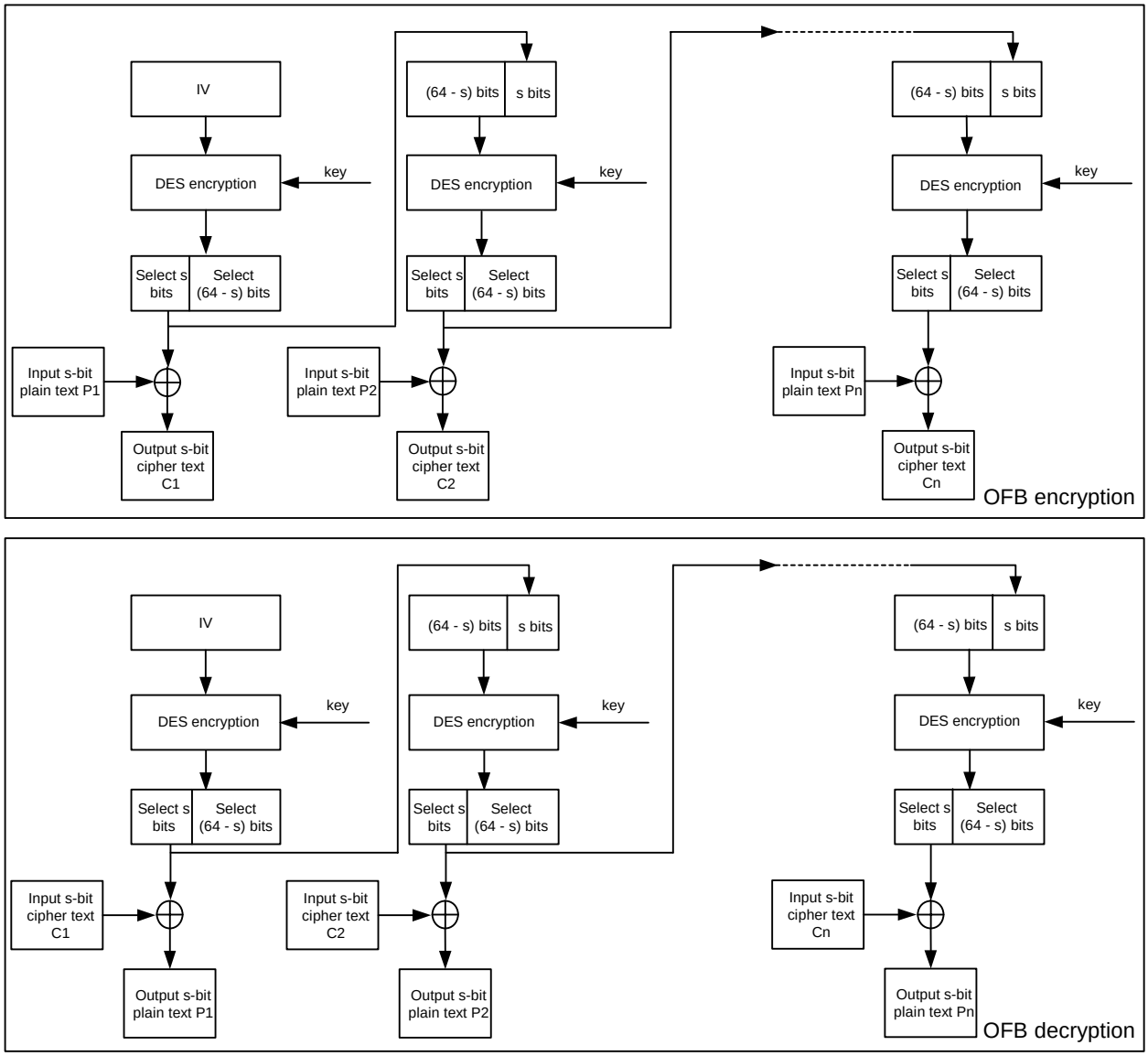
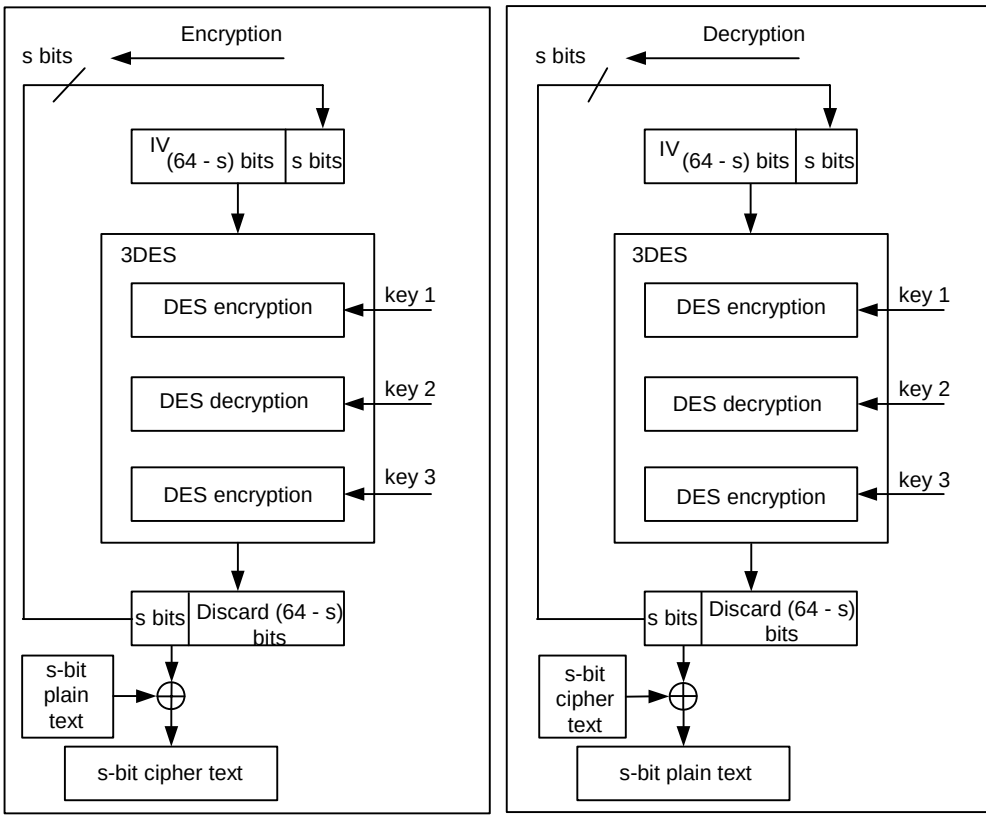


Figure 3-21 shows the s-bit OFB mode of the 3DES algorithm.

Figure 3-21 S-bit OFB mode of the 3DES algorithm



CTR Mode

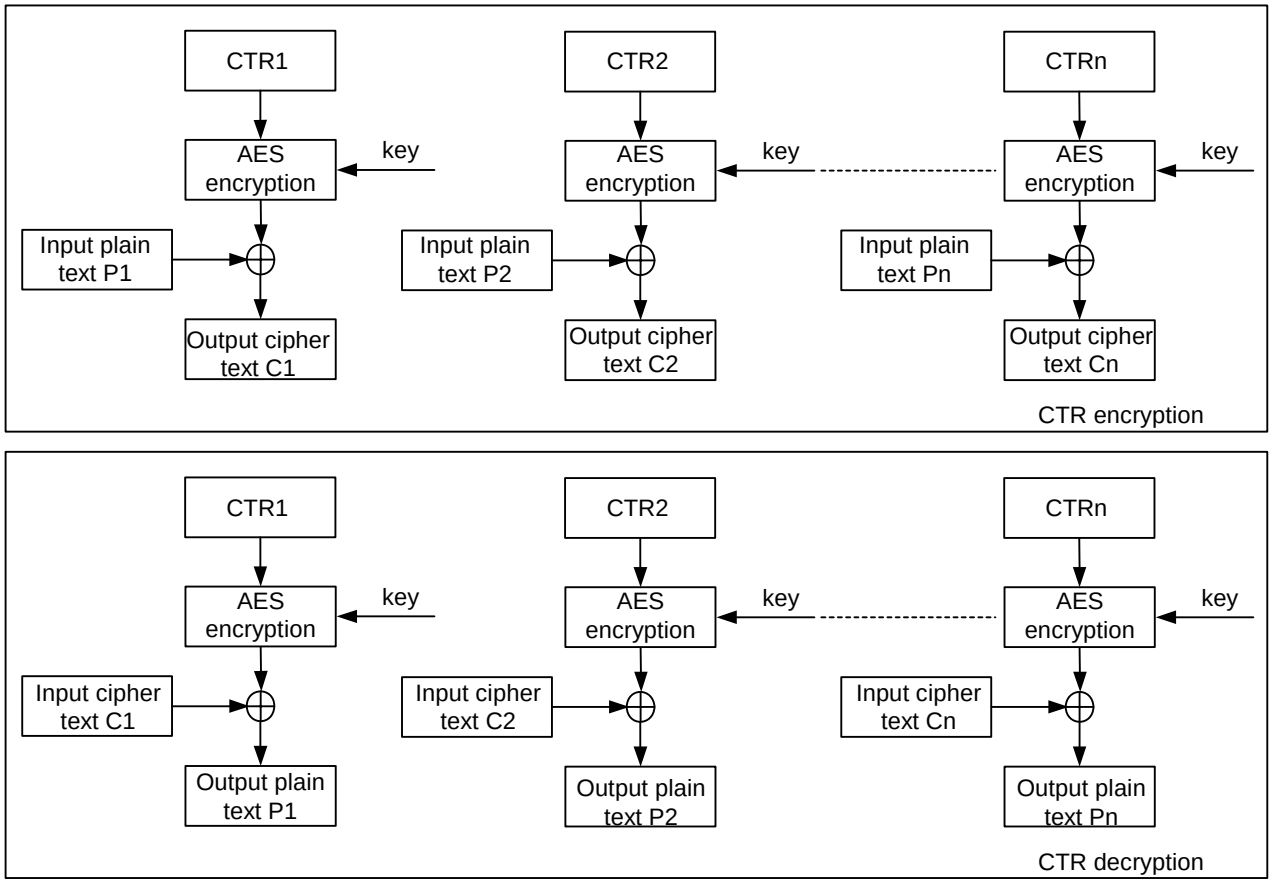
In CTR mode, different data is input to the CIPHER module by using the AES algorithm to ensure the security of data processing. Such data can be the count value CTRn. Therefore, CTRn also ensures the security of the CTR mode.

NOTE

CTRn is obtained in accumulative mode.

Figure 3-22 shows the CTR mode of the AES algorithm.

Figure 3-22 CTR mode of the AES algorithm



3.6.4 Operating Mode

Single-Block Process of the CIPHER Module

The single-block operation of the CIPHER module is performed as follows:

Step 1 Read the status register CIPHER_BUSY. If it is 0, go to Step 2; otherwise, continue to read this register.

Step 2 Configure CIPHER_CTRL.

Step 3 Configure the registers CIPHER_DIN, CIPHER_IVIN, and CIPHER_KEY.

Step 4 Set CIPHER_ST to 0x1.

Step 5 Wait until the operation is complete in either of the following mode:

- In interrupt mode, if an interrupt occurs, go to Step 6; otherwise, wait until an interrupt occurs.
- In query mode, read CIPHER_BUSY. If it is 0, go to Step 6; otherwise, continue to read this register.

Step 6 Read the result registers CIPHER_DOUT and CIPHER_IVOUT.

----End

The single-block operation is available in all the operating modes.

If you want to encrypt and decrypt a segment of data through the single-block operation, the preceding process must be performed repeatedly. To improve efficiency, note the following points:

- For the same segment of data, if the key is the same, CIPHER_KEY needs to be configured for the first block only.
- In CBC, CFB, or OFB mode, after a block operation, the obtained value of CIPHER_IVOUT is used as the value of CIPHER_IVIN in the next operation by default. Therefore, the next operation can be started in either of the following ways:
  - Set CIPHER_CTRL[ivin_sel] to 0 and then configure CIPHER_DIN.
  - Read the value of CIPHER_IVOUT in the current operation, set CIPHER_IVIN to this value, set CIPHER_CTRL[ivin_sel] to 1, and then configure CIPHER_CTRL.
- In CBC, CFB, or OFB mode, if the value of CIPHER_IVIN in the current operation is irrelevant to the value of CIPHER_IVOUT in the previous block, you must reconfigure CIPHER_IVIN and set CIPHER_CTRL[ivin_sel] to 1.

Multi-Block Process of the CIPHER Module

The multi-block process of the CIPHER module is performed as follows:

Step 1 Read CIPHER_BUSY. If it is 0, go to Step 2; otherwise, continue to read this register.

Step 2 Configure CIPHER_CTRL.

Step 3 Configure the data storage registers SRC_START_ADDR and DEST_START_ADDR and the data amount register MEM_LENGTH.

Step 4 Configure CIPHER_IVIN and CIPHER_KEY.

Step 5 Set CIPHER_ST to 0x1.

Step 6 Wait until the operation is complete in either of the following modes:

- In interrupt mode, if an interrupt occurs, go to Step 7; otherwise, wait until an interrupt occur.
- In query mode, read CIPHER_BUSY. If it is 0, go to Step 7; otherwise, continue to read this register.

Step 7 Read CIPHER_IVOUT.

----End

The multi-block operation is available in all the operating mode except the AES CTR mode.

The multi-block operation is useful for encrypting and decrypting a large amount of data. Through this operation, the CPU reduces the times of reading and writing the registers when starting and ending each block operation. After the CPU configures the required control registers, data registers, and then CIPHER_ST, the hardware performs all the operations automatically until the data operations are complete. In this way, CPU resources are released.

For the multi-block operation, the data stored in one continuous address can be encrypted and decrypted at a time. The addresses for storing the input data and result data can be different. If you want to encrypt or decrypt the data stored in multiple addresses by using a same key, note the following requirements:

- CIPHER_KEY need not to be configured.
- The information such as the source address, target address, and data length need to be reconfigured.
- After a block operation in CBC, CFB, or OFB mode, the obtained value of CIPHER_IVOUT is used as the value of CIPHER_IVIN in the next operation by default. Therefore, the next operation can be started in either of the following ways:
  - Set CIPHER_CTRL[ivin_sel] to 0 and then configure CIPHER_DIN.
  - Read the value of CIPHER_IVOUT in the current operation, set CIPHER_IVIN to this value, set CIPHER_CTRL[ivin_sel] to 1, and then configure CIPHER_DIN.

Clock Gating

When no encryption is required and CIPHER_BUSY[cipher_busy] is set to 0b0, the clock of the CIPHER module can be disabled by configuring the registers of the system controller, thus reducing power consumption.

- Write 1 to SC_PERDIS[cipherclkdis] to disable the clock of the CIPHER module.
- Write 1 to SC_PEREN[cipherclken] to enable the clock of the CIPHER module.

Soft Reset

When an error occurs in a module, the error can be rectified through soft reset by configuring the registers of the system controller.

- Write 1 to SC_PERCTRL8[cipher_srst] to soft-reset the CIPHER module.
- Write 0 to SC_PERCTRL8[cipher_srst] to clear the soft reset on the CIPHER module, and then perform normal operations.

3.6.5 Register Summary

Table 3-24 lists the CIPHER registers.

Table 3-24 Summary of the CIPHER registers (base address: 0x100C_0000)

Offset Address	Reigster	Description	Page
0x000, 0x004, 0x008, 0x00C	CIPHER_DIN	128-bit block input register	3-73
0x010, 0x014, 0x018, 0x01C	CIPHER_IVIN	Vector block input register. It need not to be configured in ECB mode.	3-74
0x020, 0x024, 0x028, 0x02C 0x030, 0x034, 0x038, 0x03C	CIPHER_KEY	Key input register	3-75
0x040, 0x044, 0x048, 0x04C	CIPHER_DOUT	128-bit block output register	3-76
0x050, 0x054, 0x058, 0x05C	CIPHER_IVOUT	Operation complete vector output register. It is ignored in ECB and CTR modes.	3-77

Offset Address	Reigster	Description	Page
0x060	CIPHER_CTRL	Control register	3-78
0x064	INT_CIPHER	Masked interrupt register	3-81
0x068	CIPHER_BUSY	Operation status indicator register	3-82
0x06C	CIPHER_ST	Operation start/end control signal register	3-82
0x070	SRC_START_ADDR	Start address register of the off-chip memory for storing the block data to be processed	3-83
0x074	MEM_LENGTH	Length register of the block data to be processed (measured by 32 bits)	3-83
0x078	DEST_START_ADDR	Start address register of the off-chip memory for storing operation results by blocks	3-85
0x07C	INT_MASK	Interrupt mask register	3-85
0x080	INT_CIPHER_STATUS	Interrupt status register	3-86

3.6.6 Register Description

CIPHER_DIN

CIPHER_DIN is the 128-bit block input register of the CIPHER module.

Note the following points when configuring this register:

If the single-block operation is selected (CIPHER_CTRL[cipher_mode] = 0b0), CIPHER_DIN must be configured.

- Assume that the AES algorithm (CIPHER_CTRL[alg_sel] = 0b10) is selected.
  - If the 1-CFB operating mode is selected (CIPHER_CTRL bit[5:1] = 0b10010), the least significant bit (LSB) is valid, that is, CIPHER_DIN bit[0] is valid.
  - If the 8-CFB operating mode is selected (CIPHER_CTRL bit[5:1] = 0b01010), eight lower bits are valid, that is, CIPHER_DIN bit[7:0] are valid.
  - If the 128-CFB operating mode is selected (CIPHER_CTRL bit[5:1] = 0b00010 or 0b11010), 128 bits are valid.
  - In other operating modes, 128 bits are valid.
- Assume that the DES or the 3DES algorithm is selected (CIPHER_CTRL[alg_sel] = 0b00, 0b01, or 0b11).
  - If the 1-CFB or 1-OFB operating mode is selected (CIPHER_CTRL bit[5:1] = 0b10010 or 0b10011), the LSB is valid, that is, CIPHER_DIN bit[0] is valid.
  - If the 8-CFB or 8-OFB operating mode is selected (CIPHER_CTRL bit[5:1] = 0b01010 or 0b01011), eight lower bits are valid, that is, CIPHER_DIN bit[7:0] are valid.

- For the 64-CFB or 64-OFB operating mode (CIPHER_CTRL bit[5:1] = 0b00010, 0b11010, 0b00011, or 0b11011), 64 lower bits are valid, that is, CIPHER_DIN bit[63:0] is valid.
- In other operating modes, 64 lower bits are valid, that is, CIPHER_DIN bit[63:0] are valid.

If the multi-block operation is selected (CIPHER_CTRL[cipher_mode] = 0b1), CIPHER_DIN need not to be configured.

Offset Address	Register Name	Total Reset Value
0x000–0x00C	CIPHER_DIN	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	cipher_din	128-bit block input of the CIPHER module. Each address maps to a 32-bit data segment. CIPHER_DIN[31:0]: 0x000 address CIPHER_DIN[63:32]: 0x004 address CIPHER_DIN[95:64]: 0x008 address CIPHER_DIN[127:96]: 0x00C address

CIPHER_IVIN

CIPHER_IVIN is the vector block input register of the CIPHER module.

Note the following points when configuring this register:

- Assume that the single-block operation (CIPHER_CTRL bit[cipher_mode] = 0b0) is selected and the selected mode (CIPHER_CTRL[mode]=0b001, 0b010, 0b011, or 0b100) is not ECB mode.
  - If you do not need to configure the input vector (CIPHER_CTRL[ivin_sel] = 0b0), CIPHER_IVIN is not required to configure.
  - If you want to configure the input vector (CIPHER_CTRL[ivin_sel] = 0b1), CIPHER_IVIN must be configured. If the AES algorithm is selected (CIPHER_CTRL [alg_sel] = 0b10), CIPHER_IVIN bit[127:0] are valid. If the DES or 3DES algorithm is selected (CIPHER_CTRL[alg_sel] = 0b00, 0b01, or 0b11), 64 lower bits are valid, that is, CIPHER_IVIN bit[63:0] are valid.
- Assume that the multi-block operation is selected (CIPHER_CTRL bit[cipher_mode] = 0b1) and the selected mode is not ECB mode (CIPHER_CTRL[mode]=0b001, 0b010, or 0b011).
  - If you do not need to configure the input vector (CIPHER_CTRL[ivin_sel] = 0b0), CIPHER_IVIN is not required to configure, because the input vector of the first block operation is unrelated to the value of CIPHER_IVIN.

– If you need to configure the input vector (CIPHER_CTRL[ivin_sel] = 0b1), CIPHER_IVIN must be configured, because the input vector of the first block operation is obtained from CIPHER_IVIN.

Offset Address	Register Name	Total Reset Value
0x010–0x01C	CIPHER_IVIN	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	cipher_ivin	128-bit IV of the CIPHER module or the data input from the counter. Each address maps to a 32-bit data segment. CIPHER_IVIN[31:0]: 0x010 address CIPHER_IVIN[63:32]: 0x014 address CIPHER_IVIN[95:64]: 0x018 address CIPHER_IVIN[127:96]: 0x01C address

CIPHER_KEY

CIPHER_KEY is the key input register of the CIPHER module.

Note the following points when configuring this register:

- Assume that the DES algorithm is selected (CIPHER_CTRL[alg_sel] = 0b00 or 0b11), the 64 lower bits are valid, that is, CIPHER_KEY[63:0] are valid.
- Assume that the 3DES algorithm is selected (CIPHER_CTRL[alg_sel] = 0b01).

  If the three-key operation is selected (CIPHER_CTRL[key_length] = 0b00, 0b01, 0b10), 192 lower bits are valid.
  - CIPHER_KEY bit[63:0] indicates the first key.
  - CIPHER_KEY bit[127:64] indicates the second key.
  - CIPHER_KEY bit[191:128] indicates the third key.

  If the two-key operation is selected (CIPHER_CTRL[key_length] = 0b11), 128 lower bits are valid.
  - CIPHER_KEY bit[63:0] indicates the first key.
  - CIPHER_KEY bit[127:64] indicates the second key.
- Assume that the AES algorithm is selected (CIPHER_CTRL[alg_sel] = 0b10).
  - If the 128-bit key operation is selected (CIPHER_CTRL[key_length] = 0b00 or 0b11), 128 lower bits are valid, that is, CIPHER_KEY bit[127:0] are valid.
  - If the 192-bit key operation is selected (CIPHER_CTRL[key_length] = 0b01), 192 lower bits are valid, that is, CIPHER_KEY bit[191:0] are valid.
  - If the 256-bit key operation is selected (CIPHER_CTRL[key_length] = 0b10), 256 lower bits are valid.

Offset Address	Register Name	Total Reset Value
0x020–0x03C	CIPHER_KEY	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	cipher_key	Key input of the CIPHER module. Each address maps to a 32-bit data segment. CIPHER_KEY[31:0]: 0x020 address CIPHER_KEY[63:32]: 0x024 address CIPHER_KEY[95:64]: 0x028 address CIPHER_KEY[127:96]: 0x02C address CIPHER_KEY[159:128]: 0x030 address CIPHER_KEY[191:160]: 0x034 address CIPHER_KEY[223:192]: 0x038 address CIPHER_KEY[255:224]: 0x03C address

CIPHER_DOUT

CIPHER_DOUT is the 128-bit block output register of the CIPHER module.

Note the following points when reading this register:

If the single-block operation is selected (CIPHER_CTRL[cipher_mode] = 0b0), the data read from CIPHER_DOUT is the result of the single-block operation. The result of the AES algorithm is different from that of the DES or 3DES algorithm.

- Assume that the AES algorithm (CIPHER_CTRL[alg_sel] = 0b10) is selected.
  - If the 1-CFB mode is selected (CIPHER_CTRL bit[5:1] = 0b10010), the LSB is valid, that is, CIPHER_DOUT bit[0] is valid.
  - If the 8-CFB mode is selected (CIPHER_CTRL bit[5:1] = 0b01010), eight lower bits are valid, that is, CIPHER_DOUT bit[7:0] are valid.
  - If the 128-CFB operating mode is selected (CIPHER_CTRL bit[5:1] = 0b00010 or 0b11010), 128 bits are valid.
  - In other operating modes, 128 bits are valid.
- Assume that the DES or the 3DES algorithm is selected (CIPHER_CTRL[alg_sel] = 0b00, 0b01, or 0b11).
  - If the 1-CFB or 1-OFB mode is selected (CIPHER_CTRL bit[5:1] = 0b10010 or 0b10011), the LSB is valid, that is, CIPHER_DOUT bit[0] is valid.
  - If the 8-CFB or 8-OFB operating mode is selected (CIPHER_CTRL bit[5:1] = 0b01010 or 0b01011), eight lower bits are valid, that is, CIPHER_DOUT bit[7:0] are valid.

- If the 64-CFB or 64-OFB operating mode is selected (CIPHER_CTRL bit[5:1] = 0b00010, 0b00011, 0b11010, or 0b11011), 64 lower bits are valid, that is, CIPHER_DOUT bit[63:0] are valid.
- In other operating modes, 64 lower bits are valid, that is, CIPHER_DOUT bit[63:0] are valid.

If the multi-block operation is selected (CIPHER_CTRL[ivin_sel] = 0b1), the data read from CIPHER_DOUT is the result of the last block operation.

Offset Address	Register Name	Total Reset Value
0x040–0x04C	CIPHER_DOUT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	cipher_dout	128-bit block output of the CIPHER module. Each address maps to a 32-bit data segment. CIPHER_DOUT[31:0]: 0x040 address CIPHER_DOUT[63:32]: 0x044 address CIPHER_DOUT[95:64]: 0x048 address CIPHER_DOUT[127:96]: 0x04C address

CIPHER_IVOUT

CIPHER_IVOUT is the operation complete vector output register of the CIPHER module.

Note the following points when reading this register:

- If the ECB or CTR operating mode is selected (CIPHER_CTRL[mode] = 0b000, 0b100, 0b101, 0b110, or 0b111), CIPHER_IVOUT can be ignored.
- Assume that the single-block operation is selected (CIPHER_CTRL[cipher_mode] = 0b0), the data of CIPHER_IVOUT is the vector output of the block that can be regarded as the vector input of the next block operation of the same data packet.
  - If the AES algorithm is selected (CIPHER_CTRL[alg_sel] = 0b10), 128 bits are valid.
  - If the DES or 3DES algorithm is selected (CIPHER_CTRL[cipher_mode] = 0b00, 0b01, or 0b11), 64 lower bits are valid, that is, CIPHER_IVOUT bit[63:0] are valid.
- Assume that the multi-block operation is selected (CIPHER_CTRL[ivin_sel] = 0b1), the data read from CIPHER_IVOUT is the vector output of the last block operation.
  - If the AES algorithm is selected (CIPHER_CTRL[cipher_mode] = 0b10), 128 bits are valid.
  - If the DES or 3DES algorithm is selected (CIPHER_CTRL[cipher_mode] = 0b00, 0b01, or 0b11), 64 lower bits are valid, that is, CIPHER_IVOUT bit[63:0] are valid.

Offset Address	Register Name	Total Reset Value
0x050–0x05C	CIPHER_IVOUT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	cipher_ivout	Vector output after the operation of the CIPHER module is complete. It can be ignored in ECB or CTR operating mode. Each address maps to a 32-bit data segment. CIPHER_IVOUT[31:0]: 0x050 address CIPHER_IVOUT[63:32]: 0x054 address CIPHER_IVOUT[95:64]: 0x058 address CIPHER_IVOUT[127:96]: 0x05C address

CIPHER_CTRL

CIPHER_CTRL is the control register of the CIPHER module.

Note the following points when configuring this register:

- Configure this register before configuring others registers of the CIPHER module.
- The multi-block operation is not supported in the CTR operating mode of the AES algorithm. That is, when the CTR mode of the AES algorithm is selected, CIPHER_CTRL[cipher_mode] cannot be set to 1.
- In other operating modes except the CFB mode of the AES algorithm, CIPHER_CTRL[width] cannot be set to 01 or 10.
- In other operating modes except the CFB and OFB modes of the DES or 3DES algorithm, CIPHER_CTRL[width] cannot be set to 01 or 10.
- CIPHER_CTRL[byte_seq_reg] and CIPHER_CTRL[byte_seq_ram] are used to adjust the byte sequences of the data in the configuration register and result register. If the data (such as the data processed through the DES algorithm) is transmitted to the CIPHER module as character streams, the byte sequences must be adjusted.

  For example, the text 7654321 maps to the number 0x3736_3534_3332_3120. After data is received as characters, the data is stored in the memory in the following sequences:

  - The data stored in the 0x0 address is 0x3435_3637.
  - The data stored in the 0x4 address is 0x2031_3233.

Assume that the data is stored starting from the offset address of 0x0. In this case, both the byte_seq_ram and byte_seq_reg bits must be set to 1.

Offset Address	Register Name	Total Reset Value
0x060	CIPHER_CTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																	byte_seq_ram	byte_seq_reg	dest_addr_set	cipher_mode	ivin_sel	alg_sel		key_length		width		mode			Decrypt
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:15]	RO	reserved	Reserved.
[14]	RW	byte_seq_ram	Controls whether to adjust the byte sequences of the data stored in the memory address space. 0: do not adjust 1: adjust
[13]	RW	byte_seq_reg	Controls whether to adjust the byte sequences of the data configured by input data registers or read from output data registers. 0: do not adjust 1: adjust The input data registers include CIPHER_KEY, CIPHER_IVIN, and CIPHER_DIN. The output data registers include CIPHER_IVOUT and CIPHER_DOUT.
[12]	RW	dest_addr_set	Controls the starting addresses of the off-chip memory for storing the block data to be processed and the block data of operation results. 0: The start addresses for storing the block data to be processed and the block data of operation results are the same, so SRC_START_ADDR and DEST_START_ADDR need not to be configured. 1: The start addresses for storing the block data to be processed and the block data of operation results are different, so SRC_START_ADDR and DEST_START_ADDR need to be configured.
[11]	RW	cipher_mode	Cipher mode selection of the CIPHER module. 0: single-block operation 1: multi-block operation
[10]	RW	ivin_sel	Input selection of CIPHER_IVIN. 0: CIPHER_IVIN need not to be configured. 1: CIPHER_IVINmust be configured.

[9:8]	RW	alg_sel	Algorithm selection. 00: DES algorithm 01: 3DES algorithm 10: AES algorithm 11: DES algorithm
[7:6]	RW	key_length	Key length selection. For the AES algorithm: 00: 128-bit key 01: 192-bit key 10: 256-bit key 11: 128-bit key For the DES algorithm: 00: 3-key 01: 3-key 10: 3-key 11: 2-key
[5:4]	RW	width	Bit width selection. For the DES or 3DES algorithm: 00: 64 bits 01: 8 bits 10: 1 bits 11: 64 bits For the AES algorithm: 00: 128 bits 01: 8 bits 10: 1 bits 11: 128 bits

[3:1]	RW	mode	Operating mode selection. For the AES algorithm: 000: ECB mode 001: CBC mode 010: CFB mode 011: OFB mode 100: CTR mode Others: ECB mode For the DES algorithm: 000: ECB mode 001: CBC mode 010: CFB mode 011: OFB mode Others: ECB mode
[0]	RW	decrypt	Encryption/decryption selection. 0: encryption 1: decryption

INT_CIPHER

INT_CIPHER is the masked interrupt register. It is the interrupt register that is generated after the interrupt status register INT_CIPHER_STATUS is masked by the interrupt mask register INT_MASK.

Offset Address	Register Name	Total Reset Value
0x064	INT_CIPHER	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1]	RO	int_error	Error interrupt status. An error occurs when the CIPHER module accesses the bus or the bus is accessed in non-word mode. 0: No error occurs. 1: An error occurs.

[0]	RO	int_done	CIPHER operation complete interrupt. 0: The operation is not complete. 1: The operation is complete.

CIPHER_BUSY

CIPHER_BUSY is the operation status indicator register.

Offset Address	Register Name	Total Reset Value
0x068	CIPHER_BUSY	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RO	cipher_busy	CIPHER operation status. 0: No CIPHER operation is performed. 1: A CIPHER operation is being performed.

CIPHER_ST

CIPHER_ST is the operation start/end control signal register of the CIPHER module.

Note the following points when configuring this register:

- If 0x0000_0001 is written to CIPHER_ST, the CIPHER module starts an operation.
- If 0x0000_0000 is written to CIPHER_ST, the CIPHER module stops running.
- In addition, when the multi-block operation is selected (CIPHER_CTRL bit[11] = 0b1), the CIPHER module stops running after the read or write operation is complete if the bus is being read or written. Otherwise, the module stops running immediately when the hardware reads the value 0x0000_0000.

Offset Address	Register Name	Total Reset Value
0x06C	CIPHER_ST	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RW	cipher_st	Operation start/end control signal of the CIPHER module. 0: An operation of the CIPHER module ends. 1: An operation of the CIPHER module starts.

SRC_START_ADDR

SRC_START_ADDR is the start address register of the off-chip memory for storing the block data to be processed.

If the multi-block operation is selected (CIPHER_CTRL[cipher_mode] = 0b1), SRC_START_ADDR must be configured before the CIPHER module is enabled.

Offset Address	Register Name	Total Reset Value
0x070	SRC_START_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	src_start_addr	Start address of the off-chip memory for storing the block data to be processed.

MEM_LENGTH

MEM_LENGTH is the length register of the block data to be processed (measured by 32 bits).

Note the following points when configuring this register:

- If the multi-block operation is selected (CIPHER_CTRL[cipher_mode] = 0b1), MEM_LENGTH must be configured before the CIPHER module is enabled.
- Assume that the AES algorithm is selected.

- If the ECB, CBC, OFB, or CTR operating mode is selected (CIPHER_CTRL[mode] = 0b000, 0b001, 0b011, 0b100, 0b101, 0b110, or 0b111), the value of mem_length must be an integer multiple of 4.
- If the 128-CFB operating mode is selected (CIPHER_CTRL[mode] = 0b010 and CIPHER_CTRL [width] = 0b00 or 0b11), the value of mem_length must be an integer multiple of 4.
- If the 8-CFB operating mode is selected (CIPHER_CTRL[mode] = 0b010 and CIPHER_CTRL[width] = 0b01), the value of mem_length is not restricted. The data must be stuffed to upper bits. The valid data is valid starting from lower bits. For example, if mem_length is set to 0x0000_0001, it indicates that the data of only one address is valid and the data is represented by data[31:0]. If only one byte of the actual data is valid, that is, data[7:0] are valid, other data is stuffed data and only eight lower bits are valid. If only two bytes are valid, that is, data[15:0] are valid, other data is stuffed data and only 16 lower bits are valid. The rest can be deduced by analogy.
- If the 1-CFB operating mode is selected (CIPHER_CTRL[mode] = 0b010 and CIPHER_CTRL[width] = 0b10), the value of mem_length is not restricted. The data, however, must be stuffed to upper bits. The valid data is valid starting from lower bits. For example, if mem_length is set to 0x0000_0001, it indicates that the data of only one address is valid and the data is represented by data[31:0]. If only one bit of the actual data is valid, that is, data[0] is valid, other data is stuffed data and only the LSB is valid. If only two bits are valid, that is, data[1:0] are valid, other data is stuffed data and only two lower bits are valid. The rest can be deduced by analogy.

- Assume that the DES or 3DES algorithm is selected.
  - If the ECB or CBC operating mode is selected (CIPHER_CTRL[mode] = 0b000, 0b001, 0b100, 0b101, 0b110, or 0b111), the value of mem_length must be an even.
  - If the 64-CFB or 64-OFB operating mode is selected (CIPHER_CTRL[mode] = 0b010 or 0b011 and CIPHER_CTRL[width] = 0b00 or 0b11), the value of mem_length must be an even.
  - If the 8-CFB or 8-OFB operating mode is selected (CIPHER_CTRL[mode] = 0b010 or 0b011 and CIPHER_CTRL[width] = 0b01), the value of mem_length is not restricted. The data, however, must be stuffed to upper bits. The valid data is valid starting from lower bits. For example, if mem_length is set to 0x0000_0001, it indicates that the data of only one address is valid and the data is represented by data[31:0]. If only one byte of the actual data is valid, that is, data[7:0] are valid, other data is stuffed data and only eight lower bits are valid. If only two bytes are valid, that is, data[15:0] are valid, other data is stuffed data and only 16 lower bits are valid. The rest can be deduced by analogy.
  - If the 1-CFB or 8-OFB operating mode is selected (CIPHER_CTRL[mode] = 0b010 or 0b011 and CIPHER_CTRL[width] = 0b10), the value of mem_length is not restricted. The data, however, must be stuffed to upper bits. The valid data is valid starting from lower bits. For example, if mem_length is set to 0x0000_0001, it indicates that the data of only one address is valid and the data is represented by data[31:0]. If only one bit of the actual data is valid, that is, data[0] is valid, other data is stuffed data and only the LSB is valid. If only two bits are valid, that is, data[1:0] are valid, other data is stuffed data and only two lower bits are valid. The rest can be deduced by analogy.

Offset Address	Register Name	Total Reset Value
0x074	MEM_LENGTH	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	mem_length	Length of the block data to be processed (measured by 32 bits).

DEST_START_ADDR

DEST_START_ADDR is the start address register of the off-chip memory for storing operation results by blocks.

Note the following points when configuring this register:

- If the start addresses for storing the block data to be processed and the block data of operation results are the same, that is CIPHER_CTRL[dest_addr_set] = 0b0, DEST_START_ADDR need not to be configured.
- If the start addresses for storing the block data to be processed and the block data of operation results are different (CIPHER_CTRL[dest_addr_set] = 0b1), DEST_START_ADDR must be configured.

Offset Address	Register Name	Total Reset Value
0x078	DEST_START_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	dest_start_addr	Start address of the off-chip memory for storing the operation results by blocks.

INT_MASK

INT_MASK is the interrupt mask register. It is used to control whether to mask the interrupt status register, thus determining whether to generate interrupts. When different values are written to INT_MASK, the following cases occur:

- After 0x0000_0003 is written to INT_MASK, the CIPHER module masks the status of INT_CIPHER_STATUS. That is, no interrupt is generated (INT_CIPHER bit[1:0] = 0b00).

- After 0x0000_0001 is written to INT_MASK, the CIPHER module masks the status of the int_done_status bit of INT_CIPHER_STATUS, so the status does not trigger any interrupt. That is, INT_CIPHER[int_done] is set to 0b0.
- After 0x0000_0002 is written to INT_CIPHER_STATUS, the CIPHER module masks the status of the int_error_status bit of INT_CIPHER_STATUS, so the status does not trigger any interrupt. That is, INT_CIPHER[int_error] is set to 0b0.

Offset Address	Register Name	Total Reset Value
0x07C	INT_MASK	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1]	RW	int_error_mask	Mask control of int_error. 0: An interrupt is generated. 1: No interrupt is generated.
[0]	RW	int_done_mask	Mask control of int_done. 0: An interrupt is generated. 1: No interrupt is generated.

INT_CIPHER_STATUS

INT_CIPHER_STATUS is the interrupt status register. Writing 1 clears this register. When different values are written to INT_CIPHER_STATUS, the following cases occur:

- After 0x0000_0003 is written to INT_CIPHER_STATUS, the CIPHER module clears the error interrupt and operation complete interrupt.
- After 0x0000_0001 is written to INT_CIPHER_STATUS, the CIPHER module clears the error interrupt.
- After 0x0000_0002 is written to INT_CIPHER_STATUS, the CIPHER module clears the operation complete interrupt.

Offset Address	Register Name	Total Reset Value
0x080	INT_CIPHER_STATUS	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1]	WC	int_error_status	Error interrupt status. An error occurs when the advanced high-performance bus (AHB) is accessed through the master interface of the CIPHER module or the slave interface is accessed in non-word mode. 0: No error occurs. 1: An error occurs.
[0]	WC	int_done_status	CIPHER operation complete interrupt status. 0: The operation is not complete. 1: The operation is complete.

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 timer module has two dual-timer modules: dual-timer0 and dual-timer1. Besides the different base addresses, the two dual-timer modules provide different functions in the system applications. The details about the four dual-timer modules are as follows:

- Dual-timer0 consists of timer0 and timer1 that share a base address and an interrupt signal.
- Dual-timer1 consists of timer2 and timer3 that share a base address and an interrupt signal.

Each dual-timer module consists of two timers with the same functions.

3.7.2 Features

Each dual-timer module has the following features:

- Provides two 16-bit or 32-bit down counters. Each counter has a programmable 8-bit prescaler.
- Provides a configurable count clock, that is, the clock can serve as the clock of the advanced peripheral bus (APB) or 3 MHz crystal oscillator 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

Typical Application

The timer module of the Hi3515 is designed for software. The two dual-timer modules of the Hi3515 provide different count clock configurations based on applications.

Function Principle

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 3), 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 Configure the SC_CTRL register of the system controller to set the reference clock of the clock enable signal of the timer.

Step 4 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

Clock Selection

Each timer has two optional count clocks. The following sections describe how to select a clock by taking timer0 as an example.

To select the APB clock as the count clock, do as follows:

Step 1 Set SC_CTRL[timeren0ov] to 1.

Step 2 Initialize the timer and start to count.

----End

To select the 3 MHz crystal oscillator clock as the count clock, do as follows:

Step 1 Set SC_CTRL[timeren0sel] to 0.

Step 2 Initialize the timer and start to count.

----End

3.7.5 Register Summary

The timer module consists of four timers and each timer involves a group of registers. The four groups of registers have the same features except that their base addresses and offset addresses are different. The details are as follows:

- TIMER0_XXXX and TIMER2_XXXX have the same base address rather than offset address.
- TIMER0_XXXX and TIMER1_XXXX have the same base address (0x2000_0000) rather than offset address; TIMER2_XXXX and TIMER3_XXXX have the same base address (0x2001_0000) rather than offset address.

Therefore, section 3.7.6 "Register Description" describes the registers by taking TIMER0_XXXX as an example.

Table 3-25 Summary of the timer registers (base addresses: 0x2000_0000 and 0x2001_0000)

Offset Address of Timer0/2	Offset Address of Timer1/3	Register	Description	Page
0x000	0x020	TIMERx_LOAD	Initial count value register	3-90
0x004	0x024	TIMERx_VALUE	Current count value register	3-91
0x008	0x028	TIMERx_CONTROL	Control register	3-92
0x00C	0x02C	TIMERx_INTCLR	Interrupt clear register	3-93
0x010	0x030	TIMERx_RIS	Raw interrupt status register	3-94
0x014	0x034	TIMERx_MIS	Masked interrupt status register	3-94
0x018	0x038	TIMERx_BGLOAD	Initial count value register in periodic mode	3-95

3.7.6 Register Description

NOTE

In TIMERx, the value of x ranges from 0 to 3.

TIMERx_LOAD

TIMERx_LOAD is the initial count value register. It is used to set the initial count value of each timer. Each timer (timer0–timer3) has one such register.

When a timer is in periodic mode and the count value reaches 0, the value of TIMERx_LOAD is reloaded to the timer. When a value is written to TIMERx_LOAD, the value of the current

counter is changed to the input value on the next rising edge of TIMCLK enabled by TIMCLKENx.

NOTE

- 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.

When a value is written to TIMERx_BGLOAD, the value of TIMERx_LOAD is overwritten, but the current count value of the timer remains unchanged.

If values are written to TIMERx_BGLOAD and TIMERx_LOAD before the rising edge of TIMCLK enabled by TIMCLKENx reaches, the value of the counter on the next rising edge is changed to the input value of TIMERx_LOAD. After that, each time when the counter reaches 0, the last values written to TIMERx_BGLOAD and TIMERx_LOAD are reloaded.

After TIMERx_BGLOAD and TIMERx_LOAD are written twice respectively, the return value after reading TIMERx_LOAD is the input value of TIMERx_BGLOAD.

Offset Address	Register Name	Total Reset Value
0x000	TIMER0_LOAD	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	timer0_load																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	timer0_load	Initial count value of timer0

TIMERx_VALUE

TIMERx_VALUE is the current count value register. It shows the current value of the counter that is decremented. Each timer (timer0–timer3) has one such register.

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	Register Name	Total Reset Value
0x004	TIMER0_VALUE	0xFFFF_FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	timer0_value																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Access	Name	Description
[31:0]	RO	timer0_value	Current count value of timer0 that is decremented.

TIMERx_CONTROL

TIMERx_CONTROL is the control register. Each timer (timer0–timer3) has one such register.

NOTE
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	Register Name	Total Reset Value
0x008	TIMER0_CONTROL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								timeren	timermode	intenable	reserved	timerpre		timersize	oneshot
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7]	RW	timeren	Timer enable. 0: disabled 1: enabled
[6]	RW	timermode	Timer count mode. 0: free-running mode 1: periodic mode
[5]	RW	intenable	Raw interrupt mask. 0: masked 1: not masked
[4]	RO	reserved	Reserved.

[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.
[1]	RW	timersize	Counter select. 0: 16-bit counter 1: 32-bit counter
[0]	RW	oneshot	Count mode select. 0: periodic mode or free-running mode 1: one-shot mode

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 (timer0–timer3) has one such register.

CAUTION

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	Register Name	Total Reset Value
0x00C	TIMER0_INTCLR	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	timer0_intclr																															
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Access	Name	Description
[31:0]	WO	timer0_intclr	Writing this register clears the output interrupt of timer0.

TIMERx_RIS

TIMERx_RIS is the raw interrupt status register. Each timer (timer0–timer3) has one such register.

Offset Address	Register Name	Total Reset Value
0x010	TIMER0_RIS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															timer0ris
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved. Writing this register has no effect and reading this register returns 0.
[0]	RO	timer0ris	Raw interrupt status of timer0. 0: No interrupt is generated. 1: An interrupt is generated.

TIMERx_MIS

TIMERx_MIS is the masked interrupt status register. Each timer (timer0–timer3) has one such register.

Offset Address	Register Name	Total Reset Value
0x014	TIMER0_MIS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																															timer0mis
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RO	timer0mis	Masked interrupt status of timer0. 0: The interrupt is invalid. 1: The interrupt is valid.

TIMERx_BGLOAD

TIMERx_BGLOAD is the initial count value register in periodic mode. Each timer (timer0–timer3) 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	Register Name	Total Reset Value
0x018	TIMER0_BGLOAD	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	timer0bgload																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	timer0bgload	Initial count value of timer0. Note: This register differs from TIMERx_LOAD. For details, see the descriptions of TIMERx_LOAD.

3.8 Watchdog

3.8.1 Overview

Within a period of time after an exception occurs in the system, the watchdog sends a reset signal to reset the entire system.

3.8.2 Features

The watchdog has the following features:

- Provides a 32-bit internal down counter. The count clock source is configurable.
- Supports the configurable timeout interval, namely, initial count value
- Locks registers to avoid any modification to them
- Supports generation of the timeout interrupt
- Supports generation of the reset signal
- Supports the debug mode

3.8.3 Signal Description

Table 3-26 Watchdog interface signal

Signal	Direction	Description	Pin
WDG_RST	O	Reset signal of the watchdog output interface, active low	WDGRST

CAUTION

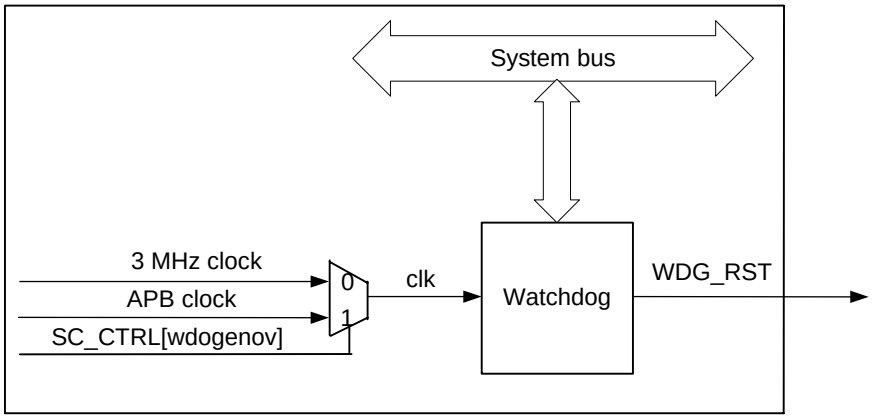
The WDGRST pin is open-drain (OD) output and it must be connected to an external pull-up resistor.

3.8.4 Function Description

Application Block Diagram

Figure 3-23 shows the application block diagram of the watchdog.

Figure 3-23 Application block diagram of the watchdog



Function Principle

The watchdog works based on a 32-bit down counter. The initial count value is loaded from 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 and the CPU does not clear the watchdog interrupt, the watchdog sends the reset signal WDG_RST and the counter stops counting.

By configuring WDG_CONTROL, you can enable or disable the watchdog to determine whether to generate an interrupt and a reset signal as required.

- When the interrupt is disabled, the counter stops counting.
- When the interrupt is enabled again, the watchdog 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.

The count clock of the watchdog can be a 3 MHz crystal oscillator clock or an APB clock, so different count time ranges are available.

By configuring WDG_LOCK, you can disable the operation of writing to the internal registers of the watchdog.

- Writing 0x1ACC_E551 to WDG_LOCK enables the write access to all the registers of the watchdog.
- Writing any other values to WDG_LOCK disables the write access to all the registers of the watchdog except WDG_LOCK.

This feature avoids improper modifications to the watchdog registers by software. Therefore, the watchdog operation is not terminated by software by mistake when an exception occurs.

In debug mode, the watchdog is disabled automatically to avoid the interference to the normal debugging.

CAUTION

The watchdog must be disabled before the system runs in sleep mode. For details, see the description of "Disabling the Watchdog" in section 3.8.5 "Operating Mode."

When either of the two processors of the Hi3515 works in debug mode, the watchdog is disabled automatically. The processor exits from the debug mode and the counter continues to count.

3.8.5 Operating Mode

Configuring the Frequency of the Count Clock

The system supports two types of watchdog count clocks: 3 MHz crystal oscillator clock and APB clock. The two clocks are selected by configuring SC_CTRL[wdogenov].

The count time of the watchdog is calculated as follows:

$$T_{WDG} = Value_{WDG\_LOAD} \times (1/f_{clk})$$

NOTE T_{WDG} indicates the count time of the watchdog, $Value_{WDG\_LOAD}$ indicates the initial count value of the watchdog, and f_{clk} indicates the frequency of the watchdog count clock.

The ranges of the count time of the watchdog in different clocks are as follows:

- When the 3 MHz crystal oscillator clock is selected, the count time ranges from 0s to 1432s.
- When the 100 MHz APB clock is selected, the count time ranges from 0s to 43s.

Initializing the Watchdog

The watchdog counter stops counting after the system is reset during power-on. When the system is initialized, the watchdog must be initialized and enabled. To initialize the watchdog, do as follows:

Step 1 Write to WDG_LOAD to set the initial count value.

Step 2 Write to WDG_CONTROL to enable the interrupt mask and the watchdog counter.

Step 3 Write to WDG_LOCK to lock the watchdog. Therefore, the settings of the watchdog are not modified by software.

----End

Processing an Interrupt

After an interrupt is received from the watchdog, the interrupt should be cleared in time and the initial count value should be reloaded to the watchdog to restart the count. To process a watchdog interrupt, do 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 the count.

Step 3 Write any value other than 0x1ACC_E551 to WDG_LOCK to lock the watchdog.

----End

Disabling the Watchdog

The watchdog must be disabled before the system runs in sleep mode. When 0 is written to WDG_CONTROL[inten], the watchdog is disabled. When 1 is written to WDG_CONTROL[inten], the watchdog is enabled.

3.8.6 Register Summary

Table 3-27 lists the watchdog registers.

Table 3-27 Summary of watchdog registers (base address: 0x2004_0000)

Offset Address	Register	Description	Page
0x000	WDG_LOAD	Initial count value register	3-99
0x004	WDG_VALUE	Current count value register	3-99
0x008	WDG_CONTROL	Control register	3-99
0x00C	WDG_INTCLR	Interrupt clear register	3-100
0x010	WDG_RIS	Raw interrupt status register	3-101
0x014	WDG_MIS	Masked interrupt status register	3-101

Offset Address	Register	Description	Page
0x018–0xBFC	RESERVED	Reserved	-
0xC00	WDG_LOCK	Lock register	3-102

3.8.7 Register Description

WDG_LOAD

WDG_LOAD is the initial count value register. It is used to configure the initial count value of the internal counter of the watchdog.

Offset Address	Register Name	Total Reset Value
0x000	WDG_LOAD	0xFFFF_FFFF

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

Bits	Access	Name	Description
[31:0]	RW	wdg_load	Initial count value of the watchdog counter.

WDG_VALUE

WDG_VALUE is the current count value register. It is used to read the current count value of the internal counter of the watchdog.

Offset Address	Register Name	Total Reset Value
0x004	WDG_VALUE	0xFFFF_FFFF

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

Bits	Access	Name	Description
[31:0]	RO	wdogvalue	Current count value of the watchdog counter.

WDG_CONTROL

WDG_CONTROL is the control register. It is used to control the functions of enabling, disabling, interrupting, and resetting the watchdog.

Offset Address	Register Name	Total Reset Value
0x008	WDG_CONTROL	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1]	RW	resen	Output enable of the watchdog reset signal. 0: disabled 1: enabled
[0]	RW	inten	Output enable of the watchdog interrupt signal. 0: The counter stops counting, the current count value keeps unchanged, and the watchdog is disabled. 1: The counter, interrupt, and watchdog are enabled. Note: If the interrupt corresponding to inten is disabled and then enabled, the counter loads the initial count value from WDG_LOAD and restarts the count.

WDG_INTCLR

WDG_INTCLR is the interrupt clear register. It is used to clear interrupts of the watchdog, thus enabling the watchdog to reload the initial count value to start the count. This register is a write-only register. The watchdog 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	Register Name	Total Reset Value
0x00C	WDG_INTCLR	-

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

Bits	Access	Name	Description
[31:0]	WO	wdg_intclr	Writing any value to this register clears the watchdog interrupts and enables the watchdog to reload the initial count value from WDG_LOAD to restart the count.

WDG_RIS

WDG_RIS is the raw interrupt register. It is used to show the status of the raw interrupts of the watchdog.

Offset Address	Register Name	Total Reset Value
0x010	WDG_RIS	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[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.

WDG_MIS

WDG_MIS is the masked interrupt status register. It is used to show the status of the masked interrupts of the watchdog.

Offset Address	Register Name	Total Reset Value
0x014	WDG_MIS	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[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.

WDG_LOCK

WDG_LOCK is the lock register. It is used to control the read and write access to watchdog registers.

Offset Address	Register Name	Total Reset Value
0xC00	WDG_LOCK	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	wdg_lock	When 0x1ACC_E551 is written to this register, the write access to all registers is enabled; when any other value is written to this register, the write access is disabled. When this register is read, the lock status other than the value written to this register is returned. 0x0000_0000: The write access is available (unlocked). 0x0000_0001: The write access is unavailable (locked).

3.9 Real Time Clock

3.9.1 Overview

The real time clock (RTC) is used to display the time in real time and send timing alarms.

3.9.2 Features

The RTC has the following features:

- Provides an internal 32-bit up counter
- Provides a count clock with the frequency of 1 Hz
- Supports the configurable initial count value
- Supports the configurable match value
- Supports the timeout interrupt
- Supports soft reset

3.9.3 Function Description

The RTC works based on a 32-bit up counter. The initial count value is loaded from RTC_LR. The count value is incremented by 1 on the rising edge of each count clock. When the count value of RTC_LR is equal to that of RTC_MR, the RTC generates an interrupt. Then the counter continues to count in incremental mode on the next rising edge of the count clock.

The interrupt number of the RTC is 10 in the system.

By configuring RTC_IMSC, you can enable or disable the RTC to determine whether to generate interrupts. At this time, there are 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_MIS shows the status of masked interrupt and RTC_RIS shows the status of raw interrupts.
- When the function of generating interrupts is enabled again, the RTC counter still counts in incremental mode. When the count value of RTC_LR is equal to that of RTC_MR, the RTC generates an interrupt.

The count clock of the RTC is a 1 Hz clock that is used to convert the count time into year, month, day, hour, minute, or second.

3.9.4 Operating Mode

Frequency of the Count Clock

The RTC uses a 1 Hz count clock. The maximum count time is calculated as follows:

$$T_{RTC} = (2^{32} - 1) \times (1/f_{rtcclk}) = 4294967295 \text{ (seconds)} \approx 49710 \text{ (days)}$$

NOTE

T_{RTC} indicates the count time, $2^{32} - 1$ indicates the initial count value, and f_{rtcclk} indicates the frequency of the count clock and it is 1 Hz.

Soft Reset

You can separately execute soft reset on the RTC by configuring SC_PERCTRL8[rtc_srst]. After soft reset, the value of each RTC configuration register is restored to its default value. Thus, these registers must be initialized again.

To execute soft reset, do as follows:

Step 1 Write 1 to SC_PERCTRL8[rtc_srst] to execute soft reset on the RTC.

Step 2 Write 0 to SC_PERCTRL8[rtc_srst] to clear the soft reset on the RTC.

----End

Initializing the RTC

The RTC counter stops counting after the system is reset during power-on. When the system is initialized, the RTC must be initialized and enabled. To initialize the RTC, do as follows:

Step 1 Set RTC_CR[rtc_start] to 0b1 to enable the RTC counter.

Step 2 Set RTC_IMSC[rtc_imsc] to 0b0 to configure the interrupt mask bit of the RTC.

Step 3 Configure RTC_MR to set the match value of the RTC.

Step 4 Configure RTC_LR to set the initial count value of the RTC.

Step 5 The RTC counts starting from the value of RTC_LR based on the frequency of the 1 Hz count clock. When the count value is equal to the value of RTC_MR, the RTC determines whether to generate interrupts based on the settings of RTC_IMSC.

----End

Processing an Interrupt

If the system receives an interrupt from the RTC, it indicates that the timing time is reached. Then, the operations such as auto startup and auto shutdown are performed. The RTC counter, however, still counts in incremental mode. To process an RTC interrupt, do as follows:

Step 1 Set RTC_ICR[rtc_icr] to 0b1 to clear the status of the RTC interrupt.

Step 2 If you want to continue to set a timing time, write a new match value to RTC_MR.

----End

Disabling the RTC

If RTC_CR is configured and the RTC starts to count, the RTC keeps on counting. The RTC can be disabled only when it is reset. For details about how to execute soft reset on the RTC, see the description of "Soft Reset" in section 3.9.4 "Operating Mode."

3.9.5 Register Summary

Table 3-28 lists the RTC registers.

Table 3-28 Summary of the RTC registers (base address: 0x2006_0000)

Offset Address	Register	Description	Page
0x000	RTC_DR	Current count value register	3-104
0x004	RTC_MR	RTC match register	3-105
0x008	RTC_LR	RTC load register	3-105
0x00C	RTC_CR	RTC enable register	3-106
0x010	RTC_IMSC	Interrupt mask register	3-106
0x014	RTC_RIS	Raw interrupt status register	3-107
0x018	RTC_MIS	Masked interrupt status register	3-107
0x01C	RTC_ICR	Interrupt clear register	3-107

3.9.6 Regitsr Description

RTC_DR

RTC_DR is the current count value register. It is used to read the current count value of the internal counter of the RTC.

Offset Address	Register Name	Total Reset Value
0x000	RTC_DR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	rtc_data	Current count value of the RTC.

RTC_MR

RTC_MR is the RTC match register. It is used to set the match value of the RTC.

Offset Address	Register Name	Total Reset Value
0x004	RTC_MR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	rtc_match	Sets the match value of the RTC.

RTC_LR

RTC_LR is the RTC load register. It is used to set the initial count value of the RTC.

Offset Address	Register Name	Total Reset Value
0x008	RTC_LR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	rtc_load	Sets the initial count value of the RTC.

RTC_CR

RTC_CR is the control register. It is used to enable the RTC. If the RTC is enabled, this register can be cleared only when the system is reset. Writing this register has no effect and reading this register returns the current value.

Offset Address	Register Name	Total Reset Value
0x00C	RTC_CR	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RW	rtc_start	RTC enable. 0: disabled 1: enabled

RTC_IMSC

RTC_IMSC is the interrupt mask register. It is used to show the interrupt mask status of the RTC.

Offset Address	Register Name	Total Reset Value
0x010	RTC_IMSC	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RW	rtc_imsc	RTC interrupt mask control. 0: masked 1: not masked

RTC_RIS

RTC_RIS is the raw interrupt status register. It is used to show the status of the raw interrupts of the RTC.

Offset Address	Register Name	Total Reset Value
0x014	RTC_RIS	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RO	rtc_ris	Status of raw interrupts of the RTC. 0: No interrupt is generated. 1: An interrupt is generated.

RTC_MIS

RTC_MIS is the masked interrupt status register. It is used to show the status of the masked interrupts of the RTC.

Offset Address	Register Name	Total Reset Value
0x018	RTC_MIS	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RO	rtc_mis	Status of the masked interrupts of the RTC. 0: No interrupt is generated or the interrupt is masked. 1: An interrupt is generated.

RTC_ICR

RTC_ICR is the RTC interrupt clear register. It is used to clear RTC interrupts.

Offset Address	Register Name	Total Reset Value
0x01C	RTC_ICR	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	WO	rtc_icr	RTC interrupt clear. 0: no effect 1: An interrupt is cleared.

3.10 System Controller

3.10.1 Overview

The system controller controls the system from various aspects. To be specific, it controls the operating mode of the system, monitors the system status, manages the important modules in the system (such as the clock module and reset module), and configures certain functions of peripherals.

3.10.2 Features

The system controller has the following features:

- Controls and monitoring the operating mode of the system.
- Controls the system clock and queries its status.
- Controls the clock gating of peripherals and supports reset control and query.
- Controls the interrupt mode of the system.
- Controls the system address remapping and monitors its status.
- Controls peripherals through their general-purpose control registers in various ways.
- Controls the arbitration function of certain buses.
- Provides write protection for key registers.
- Provides chip identification (ID) registers.

3.10.3 Function Description

Controlling the Operating Modes of the System

The system has the following four operating modes:

- Normal mode

  It refers to a mode in which the system works properly. In this mode, the system is driven by the output clock of the on-chip clock source ARMPLL. All the modules can work properly under ARMPLL.

- Slow mode

  It refers to a slow-speed mode. In this mode, the system is driven by the external crystal oscillator clock. In addition, only certain of on-chip modules can work properly, such as the system controller, timer, and SMI. None of the modules that require high-speed clock can work under the crystal oscillator clock. These modules include the DDRC, USB 2.0 HOST, AMBA-PCI bridge (the clock of the bridge is provided by the clock and reset generator), SIO0, SIO1, SIO2, and MMC.

- Doze mode

  It also refers to a slow-speed mode, but its speed is lower than that in slow mode. In this mode, the system is driven by a 46.8 kHz low-frequency clock. This clock is generated by dividing the frequency of the external crystal oscillator clock. Most on-chip peripherals and memory interfaces cannot work. Only the CPU and certain modules (such as the system controller and timer) can work properly.

- Sleep mode

  It refers to a hibernate mode. In this mode, the CPU and most modules do not work because the clock is disabled. Only the system controller and the IR module can work under the 46.8 kHz low-frequency clock. In addition, the low-frequency clock is maintained at the AHB side of the AMBA-PCI bridge.

The system controller provides a mode switching mechanism to switch the system clock source.

You can switch the system operating mode by configuring the SC_CTRL[modectrl] field. This field defines the operating mode to which the system is switched:

- 000: sleep mode
- 001: doze mode
- 01X: slow mode
- 1XX: normal mode

NOTE

The value of X can be 0 or 1.

After the system operating mode is set, the state machine switches the mode automatically without software interference. The current system status can be queried by reading SC_CTRL[modestatus]. SC_CTRL[modestatus] not only shows the preceding four main modes (normal, slow, doze, and sleep modes), but also the following intermediate states between the main modes: SW-from-PLL, SW-to-PLL, PLLCTL, SW-from-XTAL, SW-to-XTAL, and XTALCTL.

NOTE

The normal, slow, doze, and sleep modes can be switched directly. For example, if the system works in normal mode, you can switch this mode to doze mode by setting SC_CTRL [modectrl] to 001. In practice, however, the system experiences SW-from-PLL, slow, and then SW-from-XTAL mode before the system is switched to the doze mode.

After power-on reset, the system controller is in slow mode by default.

When the VIC receives interrupt inputs in interrupt mode, the target mode is specified by the interrupt response mode register rather than SC_CTRL[modectrl].

The system controller switches the system clock and system mode by working with the clock module. When the status of the state machine is changed, the system controller sends a clock switch indicator signal. Then, the clock module switches the clock and sends a switch done indicator signal to the system controller. After the system controller detects this signal, the mode switch is complete.

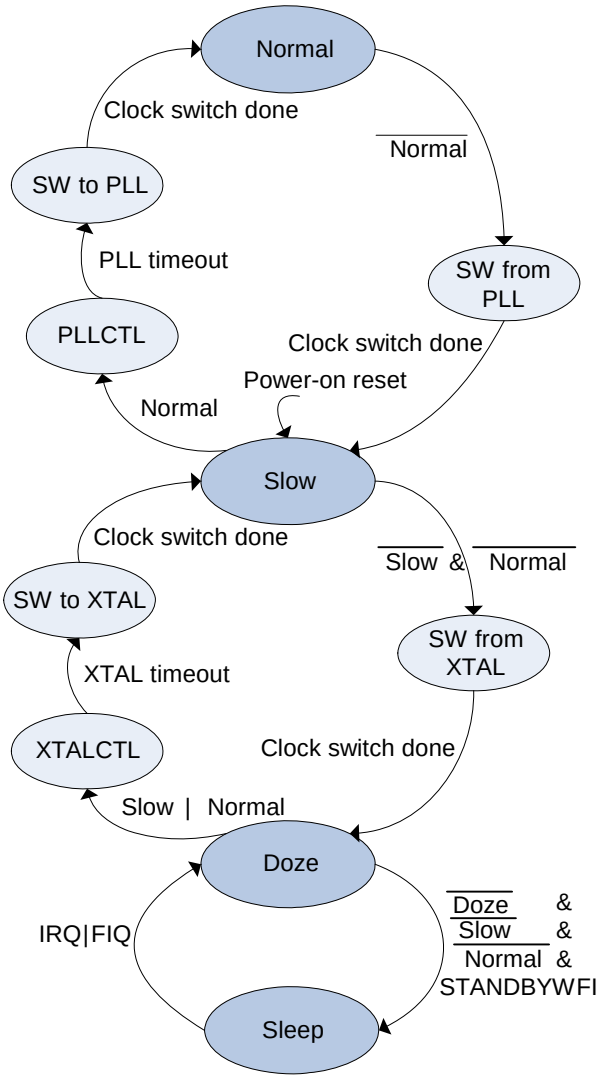
Table 3-29 lists the relationship between the status of the system controller and the system clock.

Table 3-29 Relationship between the status of the system controller and the system clock

Status of the System Controller	Status of the 24 MHz Crystal Oscillator	Status of the Master PLL	Status of the System Clock
Normal	Enabled	Enabled	The working clocks of the ARM subsystem are originated from the PLL output.
Slow	Enabled	Disabled	The working clocks of the ARM subsystem are originated from the 24 MHz crystal oscillator input.
Doze	Enabled	Disabled	The working clocks of the ARM subsystem are originated from the 46.8 kHz crystal oscillator frequency-division clock.
Sleep	Enabled	Disabled	Only the system controller and IR module work under a 46.8 kHz clock. The clocks of other modules are disabled.

Figure 3-24 shows the process of switching the system mode.

Figure 3-24 Process of switching the system mode



The operations involved in the mode switching are as follows:

- Sleep mode

  In this mode, only the clocks of the system controller and the IR module are driven by the 46.8 kHz low-speed crystal oscillator frequency-division clock. The clocks of other modules are disabled. When an FIQ or IRQ interrupt is generated, the system is switched to the doze mode and the value of SC_CTRL[modectrl] is changed from sleep to doze automatically.

NOTE

The switching from the sleep mode to the doze mode is triggered by an interrupt. Therefore, ensure that the interrupt controller does not mask the IRQ or FIQ interrupt before the system enters the sleep mode. When you want to switch the system to the doze mode by using the GPIO external input interrupt, the interrupt must be level-triggered rather than edge-triggered.

- Doze mode

  In this mode, the system clock and the clock of the system controller are driven by the 46.8 kHz crystal oscillator frequency-division clock. The doze mode is switched in the following cases:

- If SC_CTRL[modectrl] is set to slow or normal, the system enters the crystal oscillator-controlled status SC_XTALCTL and then initializes the 24 MHz crystal oscillator. After the crystal oscillator is stable, the system is switched to the SW-to-XTAL mode. After the system clock is switched from the 46.8 kHz clock to the 24 MHz clock, the system enters the slow mode.
- If SC_CTRL[modectrl] is set to sleep and the ARM926EJ-S is in wait-for-interrupt state, the system enters the sleep mode.

NOTE

- SC_XTALCTRL bit[18:3] define the stable time of the 24 MHz crystal oscillator. When the crystal oscillator is enabled, the timeout counter starts to count.
- You can enable ARM1176 to enter the low-power mode (wait-for-interrupt status) by configuring the coprocessor CP15.

- Slow mode

  In this mode, the ARM subsystem works under the 24 MHz clock. If SC_CTRL[modectrl] is set to normal, the system enters the PLL-controlled status SC_PLLCTL, and then enables the PLL. When the PLL is stable, the system enters the SW-to-PLL mode. After the system clock is switched to the PLL clock, the system enters the normal mode.

NOTE

SC_PLLCTRL bit[27:3] define the stable time of the PLL. After the PLL is enabled, the timeout counter starts to count. You can check whether the PLL is stable by querying SC_PLLCTRL23 bit[0].

If SC_CTRL[modectrl] is set to a mode in which the speed is lower than that in slow mode (such as the doze or sleep mode), the system is switched to the SW-from-XTAL mode. After the system clock is switched to the 46.8 kHz clock, the system enters the doze mode.

- Normal mode

  In this mode, the ARM subsystem works under the PLL output clock. If SC_CTRL[modectrl] is set to a mode rather than normal, the system is switched to the SW-from-PLL mode. After the system clock is switched to the 24 MHz clock, the system enters the slow mode.

PLL Control

The state machine of the system controller can enable or disable the on-chip PLL. Table 3-29 shows the PLL status in different modes.

PLL Frequency Control

The system controller is integrated with four PLL frequency control registers that are used to define the control coefficients of the PLL. For details, see the descriptions of SC_PERCTRL0 to SC_PERCTRL7.

Interrupt Response Mode

The interrupt response mode defines the mode of the system state machine after an interrupt is generated. The interrupt response mode is controlled by a group of interrupt mode control registers. These registers define the following functions:

- Enabling the interrupt response mode or not
- Operating mode of the system after an interrupt is generated
- The type of the interrupt for triggering the interrupt response mode: FIQ or IRQ

- The mechanism for querying and clearing the interrupt mode status

NOTE

In interrupt response mode, the system can switch only from a low-speed mode to a high-speed mode, such as from doze mode to normal mode. Therefore, the system cannot be switched from the normal mode to the slow mode.

Soft Reset

The system controller can execute soft reset on the entire chip or certain modules.

- After the global soft reset register SC_SYSSTAT is configured, the system controller sends a reset request to the on-chip reset module. Then the Hi3515 is reset.
- After the soft reset control bit of a module of the system controller is configured, the system controller determines whether to reset this module by controlling the on-chip reset module.

A soft reset operation involves soft reset configuration and clear. For example, to execute soft reset on UART0, you must write 1 to SC_PERCTRL8[uart0_srst], and then write 0 to SC_PERCTRL8[uart0_srst] to clear soft reset.

NOTE

- You can execute soft reset on the system by writing any value to SC_SYSSTAT. The system soft reset is cleared automatically without software interference.
- For details on how to control the soft reset on each module, see the descriptions of SC_PERCTRL8 and SC_PERCTRL10.

System Address Remapping Control

By providing address remapping control signals, the system controller allows the address decoding unit to remap and reallocate the system memory address space. After power-on reset, the Hi3515 maps address 0 to different physical spaces based on the boot mode. The Hi3515 can boot in either of the following modes by configuring the pin BOOTSEL:

- 00: booting from the NOR flash
- 00: booting from the NAND flash

NOTE

After the address remapping is cleared, the memory space mapping between address 0 and the SMI, NANDC flash is also cleared. In this case, it is recommended to allocate the address ranging from 0x0000_0000 to 0x0000_0FFF to the 2 KB on-chip ITCM by configuring the CPU.

Enable Control for WatchDog and Timer Clocks

Through the clock enable function, the count frequency can be independent of the frequency of the system clock. That is, the counter can work at a fixed count frequency even the frequency of the system clock is changed. The system controller clock provides the following enable control functions:

- Sampling input count clocks, generating clock enable signals, and then sending the signals to the watchdog and timer
- Forcibly enabling the count clocks of watchdog and timer through software. In this way, the internal counter of the watchdog and timer count based on the bus clock. When the system is in debug mode, the count function of the watchdog is disabled.
- Selecting the count clock source for the timer

Bus Arbitration Control

The 128-bit interconnect bus has an arbitration mechanism. Therefore, the system controller can set the priorities and timeout time of the ports on the interconnect bus.

Write Proctection for Key Registers

To avoid the entire system being severely affected due to the improper operation on the system controller, the system controller provides write protection for key configuration registers. Such key configuration registers are as follows:

- Mode switch control register SC_CRTL
- System global soft reset control register SC_SYSSTAT
- On-chip ARMPLL control registers SC_PERCTRL0 and SC_PERCTRL1

Before providing write protection for the preceding key registers, you must configure SC_PERLOCK to disable the write protection function. Then, you can configure SC_PERLOCK to enable the write protection function, thus ensuring that the key registers are not overwritten by software.

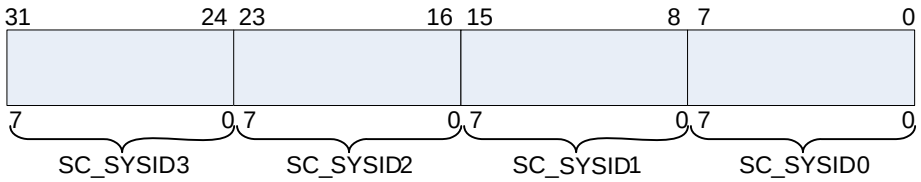
NOTE

After reset, the key registers are not write-protected by default. To enable write protection, it is recommended to configure SC_PERLOCK when the system starts.

Chip ID Registers

The system controller provides a chip ID register SC_SYSID. This register is a virtual 32-bit read-only register. In fact, it consists of four 8-bit ID registers: SC_SYSID3, SC_SYSID2, SC_SYSID1, and SC_SYSID0. After the values of these four registers are read and combined, the value of SC_SYSID, namely, 0x3515_0100, is obtained. Figure 3-25 shows the bit allocation of the chip ID registers.

Figure 3-25 Bit allocation of chip ID registers



3.10.4 Register Summary

Table 3-30 lists the registers of the system controller.

Table 3-30 Summary of the system controller registers (base address: 0x2005_0000)

Offset Address	Register	Description	Page
0x0000	SC_CTRL	System control register	3-116

Offset Address	Register	Description	Page
0x0004	SC_SYSSTAT	System status register	3-118
0x0008	SC_IMCTRL	Interrupt mode control register	3-119
0x000C	SC_IMSTAT	Interrupt mode status register	3-120
0x0010	SC_XTALCTRL	Crystal oscillator control register	3-121
0x0014	SC_PLLCTRL	PLL control register	3-121
0x001C	SC_PERCTRL0	ARMPLL frequency control register 1	3-122
0x0020	SC_PERCTRL1	ARMPLL frequency control register 2	3-123
0x0024	SC_PEREN	Peripheral clock enable register	3-124
0x0028	SC_PERDIS	Peripheral clock disable register	3-127
0x002C	SC_PERCLKEN	Peripheral clock status register	3-130
0x0034	SC_PERCTRL2	Video PLL0 frequency control register 1	3-133
0x0038	SC_PERCTRL3	Video PLL0 frequency control register 2	3-133
0x003C	SC_PERCTRL4	Video PLL1 frequency control register 1	3-134
0x0040	SC_PERCTRL5	Video PLL1 frequency control register 2	3-135
0x0044	SC_PERLOCK	Lock register of key system control registers	3-136
0x0048	SC_PERCTRL6	ETH PLL frequency control register 1	3-136
0x004C	SC_PERCTRL7	ETH PLL frequency control register 2	3-137
0x0050	SC_PERCTRL8	Soft reset control register 1	3-138
0x0054	SC_PERCTRL9	Clock mode control register 1	3-140
0x0058	SC_PERCTRL10	Soft reset control register 2	3-142
0x005C	SC_PERCTRL11	Peripheral operating mode register 1	3-144
0x0060	SC_PERCTRL12	Peripheral operating mode register 2	3-145
0x0064	SC_PERCTRL13	Clock mode control register 2	3-147
0x0068	SC_PERCTRL14	Clock mode control register 3	3-148
0x0070	SC_PERCTRL16	Clock mode control register 4	3-149
0x008C	SC_PERCTRL23	Chip operating mode status and PLL status register	3-150
0xEE0	SC_SYSID0	Chip ID register 0	3-151
0xEE4	SC_SYSID1	Chip ID register 1	3-152
0xEE8	SC_SYSID2	Chip ID register 2	3-152

Offset Address	Register	Description	Page
0xEEC	SC_SYSID3	Chip ID register 3	3-153

3.10.5 Register Description

SC_CTRL

SC_CTRL is the system control register. It is used to specify the operations that need to be performed by the system.

CAUTION

This register can be write-protected by configuring SC_PERLOCK. In addition, this register can be written only when the write protection function is disabled.

Offset Address: 0x0000 | Register Name: SC_CTRL | Total Reset Value: 0x0000_0212

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								wdogenov	timeren3ov	timeren3sel	timeren2ov	timeren2sel	timeren1ov	timeren1sel	timeren0ov	timeren0sel	reserved					remapstat	remapclear	reserved	modestatus				modectrl			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23]	RW	wdogenov	Watchdog count clock select. 0: 3 MHz clock 1: bus clock
[22]	RW	timeren3ov	Count clock select of timer3. 0: The enable signal is obtained through the reference clock. The reference clock is specified by timeren3sel. 1: bus clock
[21]	RW	timeren3sel	Count clock frequency select of timer3. This bit must be set to 0. 0: 3 MHz clock 1: reserved

[20]	RW	timeren2ov	Count clock select of timer2. 0: The enable signal is obtained through the reference clock. The reference clock is specified by timeren2sel. 1: bus clock
[19]	RW	timeren2sel	Count clock frequency select of timer2. This bit must be set to 0. 0: 3 MHz clock 1: reserved
[18]	RW	timeren1ov	Count clock select of timer1. 0: The enable signal is obtained through the reference clock. The reference clock is specified by timeren1sel. 1: bus clock
[17]	RW	timeren1sel	Count clock frequency select of timer1. This bit must be set to 0. 0: 3 MHz clock 1: reserved
[16]	RW	timeren0ov	Count clock select of timer0. 0: The enable signal is obtained through the reference clock. The reference clock is specified by timeren0sel. 1: bus clock
[15]	RW	timeren0sel	Count clock frequency select of timer0. This bit must be set to 0. 0: 3 MHz clock 1: reserved
[14:10]	RO	reserved	Reserved. Writing this bit returns 0 and reading this bit has no effect.
[9]	RO	remapstat	Status of address remapping. 0: The address is not remapped. 1: The address is remapped. When the load mode is self-load, EBICS0N is remapped to address 0; when the load mode is passive load, DDRCSN is remapped to address 0.
[8]	RW	remapclear	Address remapping clear. 0: Keep the remapping status. 1: Clear the remapping status. For details about the address mapping before and after the mapping is cleared, see section 3.3.2 "Memory Address Mapping."
[7]	RO	reserved	Reserved. Writing this bit returns 0 and reading this bit has no effect.

[6:3]	RW	modestatus	Mode status. These four bits return to the current operating mode of the system. The definitions of the four bits are as follows: 0x0: sleep 0x1: doze 0x2: slow 0x3: XTAL CTL 0x4: NORMAL 0x6: PLL CTL 0x9: SW from XTAL 0xA: SW from PLL 0xB: SW to XTAL 0xE: SW to PLL Others: reserved
[2:0]	RW	modectrl	Mode control. These three bits define the operating mode to which the system controller is switched. The definitions of the three bits are as follows: 000: sleep 001: doze 01X: slow 1XX: normal

SC_SYSSTAT

SC_SYSSTAT is the system status register. When any value is written to this register, the system controller sends a system soft reset request to the reset module. Then the reset module resets the system.

CAUTION

This register can be write-protected by configuring SC_PERLOCK. In addition, this register can be written only when the write protection function is disabled.

Offset Address	Register Name	Total Reset Value
0x0004	SC_SYSSTAT	0x0000_0002

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

Bits	Access	Name	Description
[31:0]	WO	softresreq	The system is reset when any value is written to this register.

SC_IMCTRL

SC_IMCTRL is the interrupt mode control register. It is used to control the system mode when an interrupt is generated.

Offset Address	Register Name	Total Reset Value
0x0008	SC_IMCTRL	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved. Writing this bit returns 0 and reading this bit has no effect.
[7]	RW	inmdtype	Type of the interrupt for triggering the interrupt mode of the system. 0: FIQ interrupt only 1: FIQ interrupt and IRQ interrupt
[6:4]	RO	reserved	Reserved.
[3:1]	RW	itmdctrl	Operating mode of the system with the lowest speed in interrupt mode. The value generated after the values of this register and SC_CTRL[modectrl] are ORed indicates the operating mode of the system after an interrupt is generated. The definitions of these bits are as follows: 000: sleep 001: doze 01X: slow 1XX: normal

[0]	RW	itmden	Interrupt mode enable. 0: disabled 1: enabled (when an interrupt is generated, the system enters the interrupt mode)

SC_IMSTAT

SC_IMSTAT is the interrupt mode control register. It is used to check whether the system is in interrupt mode. The system can be set to the interrupt mode forcibly by configuring this register.

CAUTION

When the ISR ends, the interrupt mode must be cleared manually.

Offset Address	Register Name	Total Reset Value
0x000C	SC_IMSTAT	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved. Writing this bit returns 0 and reading this bit has no effect.
[0]	RW	itmdstat	Interrupt mode status. The bit is used by software for controlling whether the system enters the interrupt mode. When the register is read: 0: The system is not in interrupt mode. 1: The system is in interrupt mode. When the register is written: 0: Software controls whether the system enters the interrupt mode. 1: Software does not control whether the system enters the interrupt mode.

SC_XTALCTRL

SC_XTALCTRL is the crystal oscillator control register. This register is used to control the stable time of initializing the clock module, that is, the wait time spent on switching the mode from XTAL CTL to SW-to-XTAL.

Offset Address	Register Name	Total Reset Value
0x0010	SC_XTALCTRL	0x0000_0000

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

Bits	Access	Name	Description
[31:19]	RO	reserved	Reserved. Writing these bits returns 0 and reading these bits has no effect.
[18:3]	RW	xtaltime	Wait time of crystal oscillator switching. The value of this field is used to specify the wait time spent on switching the system mode from XTAL CTL to SW-to-XTAL. The wait cycle can be calculated as follows: (65536 - xtaltime) x T46.8 K. T46.8 K indicates the clock cycle of the 46.8 kHz low-frequency clock.
[2]	RO	reserved	Reserved. Writing this bit returns 0 and reading this bit has no effect.
[1:0]	RO	reserved	Reserved. Reading these bits returns the written value.

SC_PLLCTRL

SC_PLLCTRL is the PLL control register. It is used to control whether to enable the on-chip ARMPLL through software or system mode switching. It also sets the stable time of setting the ARMPLL.

CAUTION

This register can be write-protected by configuring SC_PERLOCK. In addition, this register can be written only when the write protection function is disabled.

When the ARMPLL is enabled through the system mode switching, the ARMPLL is disabled automatically if the system is not in normal mode.

The clock frequency of the ARMPLL is controlled by certain bits of SC_PERCTRL0 and SC_PERCTRL1.

NOTE

When the PLL frequency is changed, a stable clock can be output after at least 0.5 ms. Therefore, the plltime of this register must meet this requirement.

Offset Address	Register Name	Total Reset Value
0x0014	SC_PLLCTRL	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	RO	reserved	Reserved. Writing this bit returns 0 and reading this bit has no effect.
[27:3]	RW	plltime	Stable time of the ARMPLL. This time refers to the period from the start of the PLL to the output of a stable PLL clock. That is, the wait time of switching the system mode from PLL CTL to the SW-to-PLL. The timeout time is calculated as follows: (33554432 - plltime) x TXIN. TXIN indicates the clock cycle of the external crystal oscillator of the Hi3515.
[2]	RO	reserved	Reserved. Writing this bit returns 0 and reading this bit has no effect.
[1]	RO	reserved	Reserved.
[0]	RW	pllover	This bit must be set to 0. It indicates that the ARMPLL is enabled through the system mode switching.

SC_PERCTRL0

SC_PERCTRL0 is ARMPLL frequency control register 1.

Offset Address	Register Name	Total Reset Value
0x001C	SC_PERCTRL0	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	apll_dsmpd	apll_bypass	apll_postdiv2				apll_postdiv1		apll_frac																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	apll_dsmpd	APLL frequency-division mode control. 0: decimal frequency-division mode 1: integer frequency-division mode
[30]	RW	apll_bypass	ARMPLL clock frequency-division bypass control. 0: no bypass 1: bypass
[29:27]	RW	apll_postdiv2	Level-2 output frequency divider of the APLL.
[26:24]	RW	apll_postdiv1	Level-1 output frequency divider of the APLL.
[23:0]	RW	apll_frac	Decimal frequency divider of the APLL.

SC_PERCTRL1

SC_PERCTRL1 is ARMPLL frequency control register 2.

Offset Address	Register Name	Total Reset Value
0x0020	SC_PERCTRL1	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								apll_reset	apll_pd	apll_foutvcopd	apll_postdivpd	apll_fout4phasepd	apll_refdiv							apll_fbdiv											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:23]	RO	reserved	Reserved.

[22]	RW	apll_reset	APLL reset control. 0: reset 1: not reset
[21]	RW	apll_pd	APLL power down control. 0: disabled 1: enabled
[20]	RW	apll_foutvcopd	APLL VCO output power down control. 0: disabled 1: enabled
[19]	RW	apll_postdivpd	APLL POSTDIV output power down control. 0: disabled 1: enabled
[18]	RW	apll_fout4phasepd	APLL FOUT output power down control. 0: disabled 1: enabled
[17:12]	RW	apll_refdiv	Frequency divider of the APLL reference clock.
[11:0]	RW	apll_fbdiv	Integer frequency multiplier of the APLL.

SC_PEREN

SC_PEREN is the peripheral clock enable register. As a write-only register, it generates the enable signal of the peripheral clock in the external clock generation logic. Writing 1 to a certain bit of this register enables the clock of the corresponding module, but writing 0 has no effect.

Offset Address	Register Name	Total Reset Value
0x0024	SC_PEREN	0xFFFF_FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vdac1clken	vdac0clken	sspclken	reserved	nandcclken	reserved	vohdclken	vosdclken	vobusclken	usbclken	ethclken	vi3clken	vi2clken	vi1clken	vi0clken	vibusclken	tdeclken	mmcclken	reserved		ircIken	reserved	sio1clken	sio0clken	uart3clken	uart2clken	uart1clken	uart0clken	smiclken	cipherclken	sataclkgate	reserved
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Access	Name	Description
[31]	WO	vdac1clken	Video DAC1 clock enable. 0: disabled 1: enabled

[30]	WO	vdac0clken	Video DAC0 clock enable. 0: disabled 1: enabled
[29]	WO	sspclken	SSP clock enable. 0: disabled 1: enabled
[28]	RO	reserved	Reserved.
[27]	WO	nandcclken	NANDC clock enable. 0: disabled 1: enabled
[26]	RO	reserved	Reserved.
[25]	WO	vohdclken	VO high-definition (HD) channel clock enable. 0: disabled 1: enabled
[24]	WO	vosdclken	VO standard-definition (SD) channel clock enable. 0: disabled 1: enabled
[23]	WO	vobusclken	VO bus clock enable. 0: disabled 1: enabled
[22]	WO	usbclken	USB clock enabled. 0: disabled 1: enabled
[21]	WO	ethclken	ETH clock enable. 0: disabled 1: enabled
[20]	WO	vi3clken	VI3 port clock enable. 0: disabled 1: enabled
[19]	WO	vi2clken	VI2 port clock enable. 0: disabled 1: enabled
[18]	WO	vi1clken	VI1 port clock enable. 0: disabled 1: enabled

[17]	WO	vi0clken	VI0 port clock enable. 0: disabled 1: enabled
[16]	WO	vibusclken	VI bus clock enable. 0: disabled 1: enabled
[15]	WO	tdeclken	TDE clock enable. 0: disabled 1: enabled
[14]	WO	mmcclken	MMC clock enable. 0: disabled 1: enabled
[13:12]	RO	reserved	Reserved.
[11]	WO	irclken	IR clock enable. 0: disabled 1: enabled
[10]	RO	reserved	Reserved.
[9]	WO	sio1clken	SIO1 clock enable. 0: disabled 1: enabled
[8]	WO	sio0clken	SIO0 clock enable. 0: disabled 1: enabled
[7]	WO	uart3clken	UART3 clock enable. 0: disabled 1: enabled
[6]	WO	uart2clken	UART2 clock enable. 0: disabled 1: enabled
[5]	WO	uart1clken	UART1 clock enable. 0: disabled 1: enabled
[4]	WO	uart0clken	UART0 clock enable. 0: disabled 1: enabled

[3]	WO	smiclken	SMI clock enable. 0: disabled 1: enabled
[2]	WO	cipherclken	CIPHER clock enable. 0: disabled 1: enabled
[1]	WO	sataclkgate	SATA clock enable. 0: disabled 1: enabled
[0]	RO	reserved	Reserved.

SC_PERDIS

SC_PERDIS is the peripheral clock disable register. As a write-only register, it is used to set the enable signal of the peripheral clock to be invalid in the external clock generation logic. Writing 1 to a certain bit of this register disables the clock of the corresponding module, but writing 0 has no effect.

To enable or disable the clock of a module, you must use both the SC_PEREN register and SC_PERDIS register. For example, if the clock of the SMI module is enabled, this clock is disabled when 0x0000_0004 is written to SC_PERDIS. In this case, if you want to enable this clock again, write 0x0000_0001 to SC_PEREN. In addition, you can check whether the clock of a module is enabled or disabled successfully by reading a certain bit of SC_PERCLKEN. For example, after enabling the SMI clock, you can read SC_PERCLKEN[3]. If SC_PERCLKEN[3] is 1, it indicates that the operation is successful.

Offset Address	Register Name	Total Reset Value
0x0028	SC_PERDIS	0x0400_0402

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vdac1clkdis	vdac0clkdis	sspclkdis	reserved	nandcclkdis	reserved	vohdclkdis	vosdclkdis	vobusclkdis	usbclkdis	ethclkdis	vi3clkdis	vi2clkdis	vi1clkdis	vi0clkdis	vibusclkdis	tdeclkdis	mmcclkdis	reserved		irclkdis	reserved	sio1clkdis	sio0clkdis	uart3clkdis	uart2clkdis	uart1clkdis	uart0clkdis	smiclkdis	cipherclkdis	sataclkdis	reserved
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

Bits	Access	Name	Description
[31]	WO	vdac1clkdis	Video DAC1 clock disable. 0: no effect 1: disabled
[30]	WO	vdac0clkdis	Video DAC0 clock disable. 0: no effect 1: disabled

[29]	WO	sspclkdis	SSP clock disable. 0: no effect 1: disabled
[28]	RO	reserved	Reserved.
[27]	WO	nandcclkdis	NANDC clock disable. 0: no effect 1: disabled
[26]	RO	reserved	Reserved.
[25]	WO	vohdclkdis	VO HD channel clock disable. 0: no effect 1: disabled
[24]	WO	vosdclkdis	VO SD channel clock disable. 0: no effect 1: disabled
[23]	WO	vobusclkdis	VO bus clock disable. 0: no effect 1: disabled
[22]	WO	usbclkdis	USB clock disable. 0: no effect 1: disabled
[21]	WO	ethclkdis	ETH clock disable. 0: no effect 1: disabled
[20]	WO	vi3clkdis	VI3 port clock disable. 0: no effect 1: disabled
[19]	WO	vi2clkdis	VI2 port clock disable. 0: no effect 1: disabled
[18]	WO	vi1clkdis	VI1 port clock disable. 0: no effect 1: disabled
[17]	WO	vi0clkdis	VI0 port clock disable. 0: no effect 1: disabled

[16]	WO	vibusclkdis	VI bus clock disable. 0: no effect 1: disabled
[15]	WO	tdeclkdis	TDE clock disable. 0: no effect 1: disabled
[14]	WO	mmcclkdis	MMC clock disable. 0: no effect 1: disabled
[13:12]	RO	reserved	Reserved.
[11]	WO	irclkdis	IR clock disable. 0: no effect 1: disabled
[10]	RO	reserved	Reserved.
[9]	WO	sio1clkdis	SIO1 clock disable. 0: no effect 1: disabled
[8]	WO	sio0clkdis	SIO0 clock disable. 0: no effect 1: disabled
[7]	WO	uart3clkdis	UART3 clock disable. 0: no effect 1: disabled
[6]	WO	uart2clkdis	UART2 clock disable. 0: no effect 1: disabled
[5]	WO	uart1clkdis	UART1 clock disable. 0: no effect 1: disabled
[4]	WO	uart0clkdis	UART0 clock disable. 0: no effect 1: disabled
[3]	WO	smiclkdis	SMI clock disable. 0: no effect 1: disabled

[2]	WO	cipherclkdis	CIPHER clock disable. 0: no effect 1: disabled
[1]	WO	sataclkdis	SATA clock disable. 0: no effect 1: disabled
[0]	RO	reserved	Reserved.

SC_PERCLKEN

SC_PERCLKEN is the peripheral clock status register. As a read-only register, it is used to read the enable status of each module clock to check whether the operations of writing SC_PEREN and SC_PERDIS take effect. If a bit is read as 0, it indicates that the corresponding module clock is disabled. If a bit is read as 1, it indicates that the corresponding module clock is enabled.

Offset Address	Register Name	Total Reset Value
0x002C	SC_PERCLKEN	0xEFFF_CFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vdac1clkstat	vdac0clkstat	sspclkstat	reserved	nandcclkstat	reserved	vohdclkstat	vosdclkstat	vobusclkstat	usbclkstat	ethclkstat	vi3clkstat	vi2clkstat	vi1clkstat	vi0clkstat	vibusclkstat	tdeclkstat	mmcclkstat	reserved		irclkstat	sio2clkstat	sio1clkstat	sio0clkstat	uart3clkstat	uart2clkstat	uart1clkstat	uart0clkstat	smiclkstat	cipherclkstat	sataclkstat	reserved
Reset	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Access	Name	Description
[31]	RO	vdac1clkstat	Video DAC1 clock status. 0: disabled 1: enabled
[30]	RO	vdac0clkstat	Video DAC0 clock status. 0: disabled 1: enabled
[29]	RO	sspclkstat	SSP clock status. 0: disabled 1: enabled
[28]	RO	reserved	Reserved.
[27]	RO	nandcclkstat	NANDC clock status. 0: disabled 1: enabled
[26]	RO	reserved	Reserved.

[25]	RO	vohdclkstat	Status of the VO HD channel clock. 0: disabled 1: enabled
[24]	RO	vosdclkstat	Status of the VO SD channel clock. 0: disabled 1: enabled
[23]	RO	vobusclkstat	VO bus clock status. 0: disabled 1: enabled
[22]	RO	usbclkstat	USB clock status. 0: disabled 1: enabled
[21]	RO	ethclkstat	ETH clock status. 0: disabled 1: enabled
[20]	RO	vi3clkstat	VI3 port clock status. 0: disabled 1: enabled
[19]	RO	vi2clkstat	VI2 port clock status. 0: disabled 1: enabled
[18]	RO	vi1clkstat	VI1 port clock status. 0: disabled 1: enabled
[17]	RO	vi0clkstat	VI0 port clock status. 0: disabled 1: enabled
[16]	RO	vibusclkstat	VI bus clock status. 0: disabled 1: enabled
[15]	RO	tdeclkstat	TDE clock status. 0: disabled 1: enabled
[14]	RO	mmcclkstat	MMC clock status. 0: disabled 1: enabled
[13:12]	RO	reserved	Reserved.

[11]	RO	irclkstat	IR clock status. 0: disabled 1: enabled
[10]	RO	sio2clkstat	SIO2 clock status. 0: disabled 1: enabled
[9]	RO	sio1clkstat	SIO1 clock status. 0: disabled 1: enabled
[8]	RO	sio0clkstat	SIO0 clock status. 0: disabled 1: enabled
[7]	RO	uart3clkstat	UART3 clock status. 0: disabled 1: enabled
[6]	RO	uart2clkstat	UART2 clock status. 0: disabled 1: enabled
[5]	RO	uart1clkstat	UART1 clock status. 0: disabled 1: enabled
[4]	RO	uart0clkstat	UART0 clock status. 0: disabled 1: enabled
[3]	RO	smiclkstat	SMI clock status. 0: disabled 1: enabled
[2]	RO	cipherclkstat	CIPHER clock status. 0: disabled 1: enabled
[1]	RO	sataclkstat	SATA clock status. 0: disabled 1: enabled
[0]	RO	reserved	Reserved.

SC_PERCTRL2

SC_PERCTRL2 is video PLL0 frequency control register 1.

Offset Address	Register Name	Total Reset Value
0x0034	SC_PERCTRL2	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vpll0_dsmpd	vpll0_bypass	vpll0_postdiv2				vpll0_postdiv1		vpll0_frac																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	vpll0_dsmpd	VPLL0 frequency-division mode control. 0: decimal frequency-division mode 1: integer frequency-division mode
[30]	RW	vpll0_bypass	VPLL0 clock frequency-division bypass control. 0: no bypass 1: bypass
[29:27]	RW	vpll0_postdiv2	Level-2 output frequency divider of VPLL0.
[26:24]	RW	vpll0_postdiv1	Level-1 output frequency divider of VPLL0.
[23:0]	RW	vpll0_frac	Decimal frequency divider of VPLL0.

SC_PERCTRL3

SC_PERCTRL3 is video PLL0 frequency control register 2.

Offset Address	Register Name	Total Reset Value
0x0038	SC_PERCTRL3	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved									vpll0_reset	vpll0_pd	vpll0_foutvcopd	vpll0_postdivpd	vpll0_fout4phasepd	vpll0_refdiv						vpll0_fbdiv											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:23]	RO	reserved	Reserved.

[22]	RW	vpll0_reset	VPLL0 reset control. 0: reset 1: not reset
[21]	RW	vpll0_pd	VPLL0 power down control. 0: disabled 1: enabled
[20]	RW	vpll0_foutvcopd	VPLL0 VCO output power down control. 0: disabled 1: enabled
[19]	RW	vpll0_postdivpd	VPLL0 POSTDIV output power down control. 0: disabled 1: enabled
[18]	RW	vpll0_fout4phasepd	VPLL0 FOUT output power down control. 0: disabled 1: enabled
[17:12]	RW	vpll0_refdiv	Frequency divider of the VPLL0 reference clock.
[11:0]	RW	vpll0_fbdiv	Integer frequency multiplier divider of VPLL0.

SC_PERCTRL4

SC_PERCTRL4 is video PLL1 frequency control register 1.

Offset Address	Register Name	Total Reset Value
0x003C	SC_PERCTRL4	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vpll1_dsmpd	vpll1_bypass	vpll1_postdiv2				vpll1_postdiv1		vpll1_frac																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	vpll1_dsmpd	VPLL1 frequency-division mode control. 0: decimal frequency-division mode 1: integer frequency-division mode
[30]	RW	vpll1_bypass	VPLL1 clock frequency-division bypass control. 0: no bypass 1: bypass

[29:27]	RW	vpll1_postdiv2	Level-2 output frequency divider of VPLL1.
[26:24]	RW	vpll1_postdiv1	Level-1 output frequency divider of VPLL1.
[23:0]	RW	vpll1_frac	Decimal frequency divider of VPLL1.

SC_PERCTRL5

SC_PERCTRL5 is video PLL1 frequency control register 2.

Offset Address	Register Name	Total Reset Value
0x0040	SC_PERCTRL5	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved		vpll1_reset, vpll1_pd, vpll1_foutvcopd	vpll1_postdivpd, vpll1_fout4phasepd	vpll1_refdiv	vpll1_fbdiv		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:23]	RO	reserved	Reserved.
[22]	RW	vpll1_reset	VPLL1 reset control. 0: reset 1: not reset
[21]	RW	vpll1_pd	VPLL1 power down control. 0: disabled 1: enabled
[20]	RW	vpll1_foutvcopd	VPLL1 VCO output power down control. 0: disabled 1: enabled
[19]	RW	vpll1_postdivpd	VPLL1 POSTDIV output power down control. 0: disabled 1: enabled
[18]	RW	vpll1_fout4phasepd	VPLL1 FOUT output power down control. 0: disabled 1: enabled
[17:12]	RW	vpll1_refdiv	Frequency divider of the VPLL1 reference clock.
[11:0]	RW	vpll1_fbdiv	Integer frequency multiplier of VPLL1.

SC_PERLOCK

SC_PERLOCK is the lock register of key system control registers.

Offset Address	Register Name	Total Reset Value
0x0044	SC_PERLOCK	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	scper_lockl	A register for locking the key system control registers. The key registers include SC_CTRL, SC_SYSSTAT, SC_PLLCTRL, SC_PERCTRL0, and SC_PERCTRL1. When 0x1ACC_E551 is written to this register, the write access to all registers is enabled; when any other value is written to this register, the write access is disabled. Reading this register returns the lock status rather than its written value. 0x0000_0000: The write access is available (unlocked). 0x0000_0001: The write access is unavailable (locked).

SC_PERCTRL6

SC_PERCTRL6 is ETH PLL frequency control register 1.

Offset Address	Register Name	Total Reset Value
0x0048	SC_PERCTRL6	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	epll_dsmpd	epll_bypass	epll_postdiv2			epll_postdiv1			epll_frac																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	epll_dsmpd	EPLL frequency-division mode control. 0: integer frequency-division mode 1: decimal frequency-division mode

[30]	RW	epll_bypass	EPLL clock frequency-division bypass control. 0: no bypass 1: bypass
[29:27]	RW	epll_postdiv2	Level-2 output frequency divider of EPLL.
[26:24]	RW	epll_postdiv1	Level-1 output frequency divider of the EPLL.
[23:0]	RW	epll_frac	Decimal frequency divider of the EPLL.

SC_PERCTRL7

SC_PERCTRL7 is ETH PLL frequency control register 2.

Offset Address	Register Name	Total Reset Value
0x004C	SC_PERCTRL7	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved									epll_reset	epll_pd	epll_foutvcopd	epll_postdivpd	epll_fout4phasepd	epll_refdiv						epll_fbdiv											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:23]	RO	reserved	Reserved.
[22]	RW	epll_reset	EPLL reset control. 0: reset 1: not reset
[21]	RW	epll_pd	EPLL power down control. 0: disabled 1: enabled
[20]	RW	epll_foutvcopd	EPLL VCO output power down control. 0: disabled 1: enabled
[19]	RW	epll_postdivpd	EPLL POSTDIV output power down control. 0: disabled 1: enabled
[18]	RW	epll_fout4phasepd	EPLL FOUT output power down control. 0: disabled 1: enabled

[17:12]	RW	epll_refdiv	Frequency divider of the EPLL reference clock.
[11:0]	RW	epll_fbdiv	Integer frequency multiplier of the EPLL.

SC_PERCTRL8

SC_PERCTRL8 is soft reset control register 1.

Offset Address	Register Name	Total Reset Value
0x0050	SC_PERCTRL8	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	usb_srst	usb_hrst	vohd_srst	vosd_srst	reserved	vo_srst	vi3_srst	vi2_srst	vi1_srst	vi0_srst	vi_srst	reserved		eth_srst	reserved	sio1_srst	sio0_srst	mmc_srst	reserved	i2c_srst	rtc_srst	ir_srst	uart3_srst	uart2_srst	uart1_srst	uart0_srst	cipher_srst	ssmc_srst	reserved	reserved	reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	usb_srst	USB port soft reset control. 0: soft reset 1: soft reset clear
[30]	RW	usb_hrst	USB bus soft reset control. 0: soft reset 1: soft reset clear
[29]	RW	vohd_srst	VO HD soft reset control. 0: soft reset 1: soft reset clear
[28]	RW	vosd_srst	VO SD soft reset control. 0: soft reset 1: soft reset clear
[27]	RO	reserved	Reserved.
[26]	RW	vo_srst	VO bus soft reset control. 0: soft reset 1: soft reset clear
[25]	RW	vi3_srst	VI3 port soft reset control. 0: soft reset 1: soft reset clear
[24]	RW	vi2_srst	VI2 port soft reset control. 0: soft reset 1: soft reset clear

[23]	RW	vi1_srst	VI1 port soft reset control. 0: soft reset 1: soft reset clear
[22]	RW	vi0_srst	VI0 port soft reset control. 0: soft reset 1: soft reset clear
[21]	RW	vi_srst	VI bus soft reset control. 0: soft reset 1: soft reset clear
[20:19]	RO	reserved	Reserved.
[18]	RW	eth_srst	ETH bus soft reset control. 0: soft reset 1: soft reset clear
[17]	RO	reserved	Reserved.
[16]	RW	sio1_srst	SIO1 soft reset control. 0: soft reset clear 1: soft reset
[15]	RW	sio0_srst	SIO0 soft reset control. 0: soft reset clear 1: soft reset
[14]	RW	mmc_srst	MMC soft reset control. 0: soft reset 1: soft reset clear
[13]	RO	reserved	Reserved.
[12]	RW	i2c_srst	I2C soft reset control. 0: soft reset clear 1: soft reset
[11]	RW	rtc_srst	RTC soft reset control. 0: soft reset clear 1: soft reset
[10]	RW	ir_srst	IR soft reset control. 0: soft reset clear 1: soft reset
[9]	RW	uart3_srst	UART3 soft reset control. 0: soft reset clear 1: soft reset

[8]	RW	uart2_srst	UART2 soft reset control. 0: soft reset clear 1: soft reset
[7]	RW	uart1_srst	UART1 soft reset control. 0: soft reset clear 1: soft reset
[6]	RW	uart0_srst	UART0 soft reset control. 0: soft reset clear 1: soft reset
[5]	RW	cipher_srst	CIPHER soft reset control. 0: soft reset clear 1: soft reset
[4]	RW	ssmc_srst	SSMC soft reset control. 0: soft reset clear 1: soft reset
[3]	RO	reserved	Reserved.
[2]	RO	reserved	Reserved.
[1:0]	RO	reserved	Reserved.

SC_PERCTRL9

SC_PERCTRL9 is clock mode control register 1.

Offset Address	Register Name	Total Reset Value
0x0054	SC_PERCTRL9	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						aclkout_sel	reserved			mmcsap_sel	mmcclk_sel	vo1out_sel	vo0out_sel	reserved				vi3div_sel		vi2div_sel		vi1div_sel		vi0div_sel		vi3_vi2_sel	vi1_vi0_sel	ssmcclk_sel	vo0out_clk_sel	sata_clk_sel	tde_clk_sel
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25]	RW	aclkout_sel	ACKOUT output clock select. 0: SIO0 system clock 1: SIO1 system clock

[24:23]	RO	reserved	Reserved.
[22]	RW	mmcsap_sel	Normal/reverse phase control for the MMC sampling card data clock. 0: Use the normal phase clock as the sampling clock. 1: Use the reverse phase clock as the sampling clock.
[21:20]	RW	mmcclk_sel	Frequency control for the MMC working clock. 00: 25 MHz 01: 50 MHz 10: reserved 11: 19.23 MHz
[19]	RO	vo1out_sel	Reserved.
[18]	RW	vo0out_sel	Normal/reverse phase control for the output clock of the VO0 port. 0: output normal phase clock of the VO0 port 1: output reverse phase clock of the VO0 port
[17:14]	RO	reserved	Reserved.
[13:12]	RW	vi3div_sel	Frequency-division control for the VI3 port clock. 00: VI3 port clock divided by 2 01: VI3 port clock divided by 4 10: VI3 port clock 11: reserved
[11:10]	RW	vi2div_sel	Frequency-division control for the VI2 port clock. 00: VI2 port clock divided by 2 01: VI2 port clock divided by 4 10: VI2 port clock 11: reserved
[9:8]	RW	vi1div_sel	Frequency-division control for the VI1 port clock. 00: VI1 port clock divided by 2 01: VI1 port clock divided by 4 10: VI1 port clock 11: reserved
[7:6]	RW	vi0div_sel	Frequency-division control for the VI0 port clock. 00: VI0 port clock divided by 2 01: VI0 port clock divided by 4 10: VI0 port clock 11: reserved
[5]	RW	vi3_vi2_sel	VI3 port clock select. 0: input clock of the VI3 port 1: input clock of the VI2 port

[4]	RW	vi1_vi0_sel	VI1 port clock select. 0: input clock of the VI1 port 1: input clock of the VI0 port
[3]	RW	ssmcclk_sel	Frequency ratio of the SSMC clock to the bus clock. 0: SSMC:HCLK = 1:1 1: SSMC:HCLK = 1:2
[2]	RW	vo0out_clk_sel	Output clock select of the VO0 port (for test). 0: VO0 output clock 1: VO1 output clock
[1]	RW	sata_clk_sel	SATA clock select. 0: 25 MHz clock 1: 125 MHz EPLL clock
[0]	RW	tde_clk_sel	TDE clock select. 0: 270 MHz VPLL clock 1: 266 MHz APLL clock

SC_PERCTRL10

SC_PERCTRL10 is soft reset control register 2.

Offset Address	Register Name	Total Reset Value
0x0058	SC_PERCTRL10	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								sata_alive_rst	sata_rx1_rst	sata_rx0_rst	sata_tx1_rst	sata_tx0_rst	sata_phyrst	sata_hrst	sata_rst	vedu0_stdmod_p1	vedu0_mdu_en	vedu0_vpp_en	vedu0_vedsel	vedu0_stdmod	vedu0_power_mode	reserved				dma_hrst	ddr_hrst	spi_hrst	reserved	nadc_hrst	tde_srst	tde_hrst
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:25]	RO	reserved	Reserved.
[24]	RW	sata_alive_rst	Soft reset control for the SATA controller alive clock domain. 0: soft reset 1: soft reset clear
[23]	RW	sata_rx1_rst	Soft reset control for the SATA controller rx1 clock domain. 0: soft reset 1: soft reset clear

[22]	RW	sata_rx0_rst	Soft reset control for the SATA controller rx0 clock domain. 0: soft reset 1: soft reset clear
[21]	RW	sata_tx1_rst	Soft reset control for the SATA controller tx1 clock domain. 0: soft reset 1: soft reset clear
[20]	RW	sata_tx0_rst	Soft reset control for the SATA controller tx0 clock domain. 0: soft reset 1: soft reset clear
[19]	RW	sata_phyrst	Soft reset control for SATA PHY. 0: soft reset 1: soft reset clear
[18]	RW	sata_hrst	Soft reset control for the SATA controller bus. 0: soft reset 1: soft reset clear
[17]	RW	sata_rst	Soft reset control for the SATA controller interface. 0: soft reset 1: soft reset clear
[16:11]	RW	vedu0_stdmod_p1	Reserved.
[10:7]	RO	reserved	Reserved.
[6]	RW	dma_hrst	Soft reset control for the DMA bus. 0: soft reset clear 1: soft reset
[5]	RW	ddr_hrst	Soft reset control for the DDR bus. 0: soft reset clear 1: soft reset
[4]	RW	spi_hrst	Soft reset control for the SPI bus. 0: soft reset clear 1: soft reset
[3]	RO	reserved	Reserved.
[2]	RW	nadc_hrst	Soft reset control for the NANDC bus. 0: soft reset clear 1: soft reset
[1]	RW	tde_srst	Soft reset control for the TDE. 0: soft reset clear 1: soft reset

[0]	RW	tde_hrst	Soft reset control for the TDE bus. 0: soft reset clear 1: soft reset

SC_PERCTRL11

SC_PERCTRL11 is peripheral operating mode register 1.

Offset Address	Register Name	Total Reset Value
0x005C	SC_PERCTRL11	0x0100_4000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	uart1_rtsmode	spi_port	smi_canclewait	ebinandc_time_out									ebismi_time_out												ebi_arb_delay	ebi_mormal_mode	i2c_delay_bypass	reserved				
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	uart1_rtsmode	UART1 request-to-send (RTS) signal mode. 0: normal mode 1: reverse mode
[30]	RW	spi_port	SPI CS select. 0: SPI_CS0 1: SPI_CS1
[29]	RW	smi_canclewait	External wait interrupt signal of the SMI module. 0: wait in normal mode 1: interrupt the operations performed through the SMI interface forcibly
[28:19]	RW	ebinandc_time_out	Timeout value set by the NANDC module through the EBI interface. If the access operation of the NANDC module is not responded during the timeout period, the NANDC module schedules the shared channel by taking priority over the SMI module. It is recommended to set the value of these bits to 0x020 if arbitration occurs.
[18:9]	RW	ebismi_time_out	Timeout value set by the SMI module through the EBI interface. If the access operation of the SMI module is not responded during the timeout period, the SMI module schedules the shared channel by taking priority over the NANDC module. It is recommended to set the value of these bits to 0x020 if arbitration occurs.

[8:7]	RW	ebi_arb_delay	EBI delay of the switching between the SMI and NANDC. 00: no delay 01: delay one cycle 10: delay two cycles 11: delay three cycles
[6]	RW	ebi_mormal_mode	EBI arbitration mode. 0: normal mode 1: dock mode. It is idle by default.
[5]	RW	i2c_delay_bypass	I^2C serial data (SDA) delay relative to the serial clock (SCL). 0: no delay 1: delay 300 ns
[4:0]	RO	reserved	Reserved.

SC_PERCTRL12

SC_PERCTRL12 is peripheral operating mode register 2.

Offset Address	Register Name	Total Reset Value
0x0060	SC_PERCTRL12	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												usb_start_clk	usb_susp_lgcy	reserved				usb_tune1		usb_tune0		dac1_powredown	dac0_powredown	vou_test_en	reserved				sio1_master	sio0_xfs	sio0_xck	sio0_master
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:20]	RO	reserved	Reserved.
[19]	RW	usb_start_clk	OHCI clock control signal. When the OHCI clock is suspended, this bit should be set to 1 to enable the 12 MHz and 48 MHz clocks of the OHCI module. After the two clocks are enabled, this bit should be set to 0 before the OHCI clock is suspended again. 0: After the OHCI clock is suspended, the 12 MHz and 48 MHz clocks are disabled. 1: After the OHCI clock is suspended, the 12 MHz and 48 MHz clocks are enabled.

[18]	RW	usb_susp_lgcy	Strap signal of the OHCI clock. 0: When the USB ports use the OHCI controller, utmi_suspend_o_n = 0 indicates that all the USB ports that use the OHCI controller are suspended or the OHCI controller is in the global suspend state; utmi_suspend_o_n = 1 indicates that part of the USB ports that use the OHCI controller are suspended or the OHCI controller is not in the global suspend state. 1: When USB ports use the OHCI controller, the utmi_suspend_o_n signal shows the suspend state of the corresponding port.
[17:14]	RO	reserved	Reserved.
[13:12]	RW	usb_tune1	High-speed adjust signal of the transmitter of USB port 1. The static signal is used to adjust the current at high speed. 00: The default value is decreased by 4.5%. 01: default rated operating voltage 10: The default value is increased by 4.5%. 11: The default value is increased by 9%. Note: Set this signal only when the USB PHY is reset. This value must be maintained in normal mode.
[11:10]	RW	usb_tune0	High-speed adjust signal of the transmitter of USB port 0. The static signal is used to adjust the current at high speed. 00: The default value is decreased by 4.5%. 01: default rated operating voltage 10: The default value is increased by 4.5%. 11: The default value is increased by 9%. Note: Set this signal only when the USB PHY is reset. This value must be maintained in normal mode.
[9]	RW	dac1_powredown	DAC1 power down control. 0: power down 1: power on
[8]	RW	dac0_powredown	DAC0 power down control. 0: power down 1: power on
[7]	RW	vou_test_en	VOU test enable. 0: normal mode 1: test mode
[6:4]	RO	reserved	Reserved.
[3]	RW	sio1_master	SIO1 master/slave mode. 0: slave mode. Clocks and sync signals are obtained from pins. 1: master mode. Clock and sync signals are generated in the Hi3515.

[2]	RW	sio0_xfs	SIO0 transmit frame sync signal select. 0: The SIO0 transmit channel uses the signals passing through the SIO0XFS pin or generated by the Hi3515. 1: The SIO0 transmit channel uses the signals generated by the SIO0RFS pin or the Hi3515. The SIO0XFS pin is invalid.
[1]	RW	sio0_xck	SIO0 transmit clock select. 0: The SIO0 transmit channel uses the signals passing through the SIO0XCK pin or generated by the Hi3515. 1: The SIO0 transmit channel uses the signals passing through the SIO0RCK pin or generated by the Hi3515. The SIO0XCK pin is invalid.
[0]	RW	sio0_master	SIO0 master/slave mode. 0: slave mode. Clocks and sync signals are obtained from pins. 1: master mode. Clock and sync signals are generated in the Hi3515.

SC_PERCTRL13

SC_PERCTRL13 is clock mode control register 2.

Offset Address	Register Name	Total Reset Value
0x0064	SC_PERCTRL13	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	sio0_lrclk_sel	sio0_bclk_sel	sio0clk_sel					
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:28]	RW	sio0_lrclk_sel	SIO0 sampling clock frequency control. These bits provide the options of dividing the frequency of the SIO bit stream clock by 2, 4, 8, 16, 32, 48, 64, 128, or 256. 0x0: divided by 2 0x1: divided by 4 0x2: divided by 8 0x3: divided by 16 0x4: divided by 32 0x5: divided by 48 0x6: divided by 64 0x7: divided by 128 0x8: divided by 256 Others: reserved

[27:24]	RW	sio0_bclk_sel	SIO0 bit stream clock frequency control. These bits provide the options of dividing the frequency of the SIO0 system clock by 1, 2, 3, 4, 6, 8, 12, 16, 24, 48, 64, or 128. 0x0: divided by 1 0x1: divided by 3 0x2: divided by 2 0x3: divided by 4 0x4: divided by 6 0x5: divided by 8 0x6: divided by 12 0x7: divided by 16 0x8: divided by 24 0x9: divided by 48 0x10: divided by 64 0x11: divided by 128 Others: reserved
[23:0]	RW	sio0clk_sel	SIO system clock frequency control. These bits provide the options of dividing the frequency of the EPLL output clock (500 MHz) by any value. Fsio = (sio0clk_sel x Fepll)/2^27

SC_PERCTRL14

SC_PERCTRL14 is clock mode control register 3.

Offset Address	Register Name	Total Reset Value
0x0068	SC_PERCTRL14	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				sio1_bclk_sel				sio12clk_sel																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:28]	RO	reserved	Reserved.

[27:24]	RW	sio1_bclk_sel	SIO1 bit stream clock frequency control. These bits provide the options of dividing the frequency of the SIO1 system clock by 1, 2, 3, 4, 6, 8, 12, 16, 24, 48, 64, or 128. 0x0: divided by 1 0x1: divided by 3 0x2: divided by 2 0x3: divided by 4 0x4: divided by 6 0x5: divided by 8 0x6: divided by 12 0x7: divided by 16 0x8: divided by 24 0x9: divided by 48 0x10: divided by 64 0x11: divided by 128 Others: reserved
[23:0]	RW	sio12clk_sel	SIO1 or SIO2 system clock frequency control. These bits provide options of dividing the frequency of the EPLL output clock (500 MHz) by any value. Fsio = (sio12clk_sel x Fepll)/2^27

SC_PERCTRL16

SC_PERCTRL16 is clock mode control register 4.

Offset Address	Register Name	Total Reset Value
0x0070	SC_PERCTRL16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																						sio1_bclk_edge	sio0_bclk_edge	reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:10]	RO	reserved	Reserved.
[9]	RW	sio1_bclk_edge	Normal/reverse edge output select of SIO1BCLK. 0: normal edge 1: reverse edge

[8]	RW	sio0_bclk_edge	Normal/reverse edge output select of SIO0BCLK. 0: normal edge 1: reverse edge
[7:0]	RO	reserved	Reserved.

SC_PERCTRL23

SC_PERCTRL23 is the chip operating mode status and PLL status register.

Offset Address	Register Name	Total Reset Value
0x008C	SC_PERCTRL23	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																	nf_ecc_type			nf_addr_num				nf_page_size				debug_sel	reserved	boot_mode	reserved	reserved	epll_lock		vpll1_lock		vpll0_lock		apll_lock	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:15]	RO	reserved	Reserved.
[14:13]	RO	nf_ecc_type	ECC mode when the system boots from the NAND flash. bit[14] and bit[13] map to the power-on status of the multiplexed pins NFECC0 and NFECC1 respectively. The definitions of NFECC0 and NFECC1 are as follows: 00: disabled 01: 1-bit mode 10: 4 bit mode 11: 8-bit mode
[12:11]	RO	nf_addr_num	Number of addresses when the system boots from the NAND flash. bit[12] and bit[11] map to the power-on status of the multiplexed pins NFNUM0 and NFNUM1 respectively. The definitions of NFNUM0 and NFNUM1 are as follows: 00: 3 address cycles 01: 4 address cycles 10: 5 address cycles 11: 6 address cycles

[10:9]	RO	nf_page_size	Page size when the system boots from the NAND flash. bit[10] and bit[9] map to the power-on status of the multiplexed pins NFPAGE0 and NFPAGE1 respectively. The definitions of NFPAGE0 and NFPAGE1 are as follows: 01: 2 KB 10: 4 KB Others: reserved
[8]	RO	debug_sel	Selected ARM debug mode. 0: debug ARM926 1: debug SATA PHY
[7]	RO	reserved	Reserved, fixed at 0x0 in read.
[6]	RO	boot_mode	Selected boot mode of the chip. 00: boot from the NOR flash 00: boot from the NAND flash
[5]	RO	reserved	Reserved, fixed at 0x0 in read.
[4]	RO	reserved	Reserved.
[3]	RO	epll_lock	Lock status of the Ethernet PLL. 0: unlocked 0: locked
[2]	RO	vpll1_lock	Lock status of the video1 PLL. 0: unlocked 0: locked
[1]	RO	vpll0_lock	Lock status of the video0 PLL. 0: unlocked 0: locked
[0]	RO	apll_lock	Lock status of the ARM PLL. 0: unlocked 0: locked

SC_SYSID0

SC_SYSID0 is chip ID register 0.

Offset Address	Register Name	Total Reset Value
0xEE0	SC_SYSID0	0x00

Bit	7	6	5	4	3	2	1	0
Name	sysid0							
Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RO	sysid0	Reading this register returns 0x00.

SC_SYSID1

SC_SYSID1 is chip ID register 1.

Offset Address	Register Name	Total Reset Value
0xEE4	SC_SYSID1	0x01

Bit	7	6	5	4	3	2	1	0
Name	sysid1							
Reset	0	0	0	0	0	0	0	1

Bits	Access	Name	Description
[7:0]	RO	sysid1	Reading this register returns 0x01.

SC_SYSID2

SC_SYSID2 is chip ID register 2.

Offset Address	Register Name	Total Reset Value
0xEE8	SC_SYSID2	0x15

Bit	7	6	5	4	3	2	1	0
Name	sysid2							
Reset	0	0	0	1	0	1	0	1

Bits	Access	Name	Description
[7:0]	RO	sysid2	Reading this register returns 0x15.

SC_SYSID3

SC_SYSID3 is chip ID register 3.

Offset Address	Register Name	Total Reset Value
0xEEC	SC_SYSID3	0x35

Bit	7	6	5	4	3	2	1	0
Name	sysid3							
Reset	0	0	1	1	0	1	0	1

Bits	Access	Name	Description
[7:0]	RO	sysid3	Reading this register returns 0x35.

3.11 Power Management and Low-Power Mode Control

3.11.1 Overview

In low-power mode, the power consumption of chip is reduced effectively. The Hi3515 dynamically reduces its power consumption in the following low-power control modes:

- System operating mode control

  In each operating mode except the normal mode, the power consumption is reduced to some extent. You can select different operating modes according to the actual power consumption and function requirements.
- Clock gating and clock frequency adjustment

  The Hi3515 supports the disabling of clocks. That is, you can disable the clocks of unused modules, thus reducing the power consumption. 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 dynamically reduce the power consumption of the Hi3515.
- DDR low-power control

  You can enable the self-refresh mode of the DDR to reduce the power consumption of the Hi3515.

3.11.2 Operating Modes of the System

The system provides four operating modes. For details, see "Controlling the Operating Modes of the System" in section 3.10.3 "Function Description."

3.11.3 Clock Gating and Clock Frequency Adjustment

The system provides the clock gating functions for the following modules. When a module is idle, its clock can be disabled to reduce the power consumption of the Hi3515. For details about the process, see the description in the section of "clock gating" of the following modules:

- VEDU
- MMC
- VIU
- VOU
- TDE
- CIPHER
- ETH
- SIO
- UART
- SMI
- NANDC
- USB 2.0 HOST
- SSP
- SATA
- IR

In normal mode, the system can reduce the power consumption of the Hi3515 by adjusting its operating frequency. To adjust the system operating frequency, the following steps take place:

Step 1 Disable the service module to prevent it from accessing the DDR.

Step 2 The system runs in the flash or the TCM.

Step 3 Set DDRC_DLL_CONFIG[dll_cali_en] to 1 to recalibrate the delay locked loop (DLL).

Step 4 Set DDRC_CTRL[sr_req] to 1 to request to enter the self-refresh mode.

Step 5 Query the DDRC_STATUS[in_sr] bit. When it is 1, go to Step 6.

Step 6 Configure the value of SC_PLLCTRL[27:3] as the stable time of the PLL.

Step 7 Configure SC_PERCTRL0 and SC_PERCTRL1 to control the frequency-division ratio of the PLL and switch the PLL clock. That is, output a PLL clock to the DDRC when the clock is stable.

Step 8 Set DDRC_CTRL[sr_req] to 0 to request to exit the self-refresh mode.

Step 9 Query the DDRC_STATUS[in_sr] bit until it is 0.

Step 10 Set DDRC_DLL_CONFIG[dll_cali_en] to 0 to disable the recalibration function of the DLL.

Step 11 Run the software program in the DDR and enable the service module.

----End

Besides the preceding method, you can also adjust the operating frequencies of certain modules, thus further reducing the power consumption of the system. For details, see the section "Clock Configuration" of the SMI and MMC modules.

3.11.4 DDR Low-Power Control

By using this control mode, you can dynamically control the power consumption of the pins in the DDR controller and chip of the peer DDR.

- By configuring DDRC_CONFIG[pd_en] and DDRC_CONFIG[pd_prd], you can enable the DDR to enter the low-power mode. When the bus does not access the DDR, the DDRC enables the DDR to enter the low-power mode, thus reducing the power consumption.
- By setting DDRC_CTRL[sr_req] to 1, you can enable the DDR to enter the self-refresh mode to reduce the power consumption. To enable the DDR to enter the self-refresh mode, the following requirements must be met: before the DDR enters the self-refresh mode, the program must run in the on-chip program memory or the flash memory. Additionally, the DDR self-refresh function is enabled by the DDRC.

Contents

Figures

Tables

4 Memory Controller

4.1 DDR Controller

4.1.1 Overview

The DDR2 SDRAM controller (DDRC) provides DDR2 interfaces for accessing the DDR2 SDRAM. The Hi3515 has one DDRC interface.

4.1.2 Features

The DDRC has the following features:

- Provides a DDR2 SDRAM chip select space that supports the 32-bit mode.
- Supports a maximum of 512 MB storage space (for the 32-bit DDRC).
- Supports the burst-of-four transfer mode of the DDR2 SDRAM
- Supports configurable timing parameters, thus satisfying various frequency requirements of components
- Controls the auto-refresh and self-refresh modes of the DDR SDRAM and DDR2 SDRAM
- Supports the low-power mode.
- Supports the 200 MHz operating frequency of the DDR2 SDRAM

4.1.3 Signal Description

Table 4-1 describes the signals of the DDRC interface.

Table 4-1 Signals of the DDRC interface

Signal	Direction	Description	Pin
DDRCKP0	O	Positive differential clock 0	DDRCKP0
DDRCKN0	O	Negative differential clock 0	DDRCKN0
DDRCKP1	O	Positive differential clock 1	DDRCKP1
DDRCKN1	O	Negative differential clock 1	DDRCKN1
DDRCKE	O	DDR2 SDRAM clock enable	DDRCKE

Signal	Direction	Description	Pin
DDRCSN	O	DDR2 SDRAM CS	DDRCSN
DDRRASN	O	DDR2 SDRAM RAS	DDRRASN
DDRCASN	O	DDR2 SDRAM CAS	DDRCASN
DDRWEN	O	DDR2 SDRAM WE	DDRWEN
DDRODT	O	DDR2 SDRAM ODT enable	DDRODT
DDRBA0	O	DDR2 SDRAM bank address 0	DDRBA0
DDRBA1	O	DDR2 SDRAM bank address 1	DDRBA1
DDRBA2	O	DDR2 SDRAM bank address 2	DDRBA2
DDRADR[13:0]	O	DDR SDRAM address signal	DDRADR13–DDRADR0
DDRDM0	O	DDR2 SDARM0 lower-bit DM signal, corresponding to data buses DQ0–DQ7	DDRDM0
DDRDM1	O	DDR2 SDARM0 upper-bit DM signal, corresponding to data buses DQ8–DQ15	DDRDM1
DDRDM2	O	DDR2 SDARM1 lower-bit DM signal, corresponding to data buses DQ16–DQ23	DDRDM2
DDRDM3	O	DDR2 SDARM1 upper-bit DM signal, corresponding to data buses DQ24–DQ31	DDRDM3
DDRDQSP0	I/O	Positive DQS0, corresponding to data buses DQ0–DQ7	DDRDQSP0
DDRDQSN0	I/O	Negative DQS0, corresponding to data buses DQ0–DQ7	DDRDQSN0
DDRDQSP1	I/O	Positive DQS1, corresponding to data buses DQ8–DQ15	DDRDQSP1
DDRDQSN1	I/O	Negative DQS1, corresponding to data buses DQ8–DQ15	DDRDQSN1
DDRDQSP2	I/O	Positive DQS2, corresponding to data buses DQ16–DQ23	DDRDQSP2

Signal	Direction	Description	Pin
DDRDQSN2	I/O	Negative DQS2, corresponding to data buses DQ16–DQ23	DDRDQSN2
DDRDQSP3	I/O	Positive DQS3, corresponding to data buses DQ24–DQ31	DDRDQSP3
DDRDQSN3	I/O	Negative DQS3, corresponding to data buses DQ24–DQ31	DDRDQSN3
DDRDQ[31:0]	I/O	Data bus	DDRDQ31–DDRDQ0

NOTE

- CS = chip select
- RAS = row address select
- CAS = column address select
- WE = write enable
- DQS = data strobe
- DM = data mask
- ODT = on-die termination

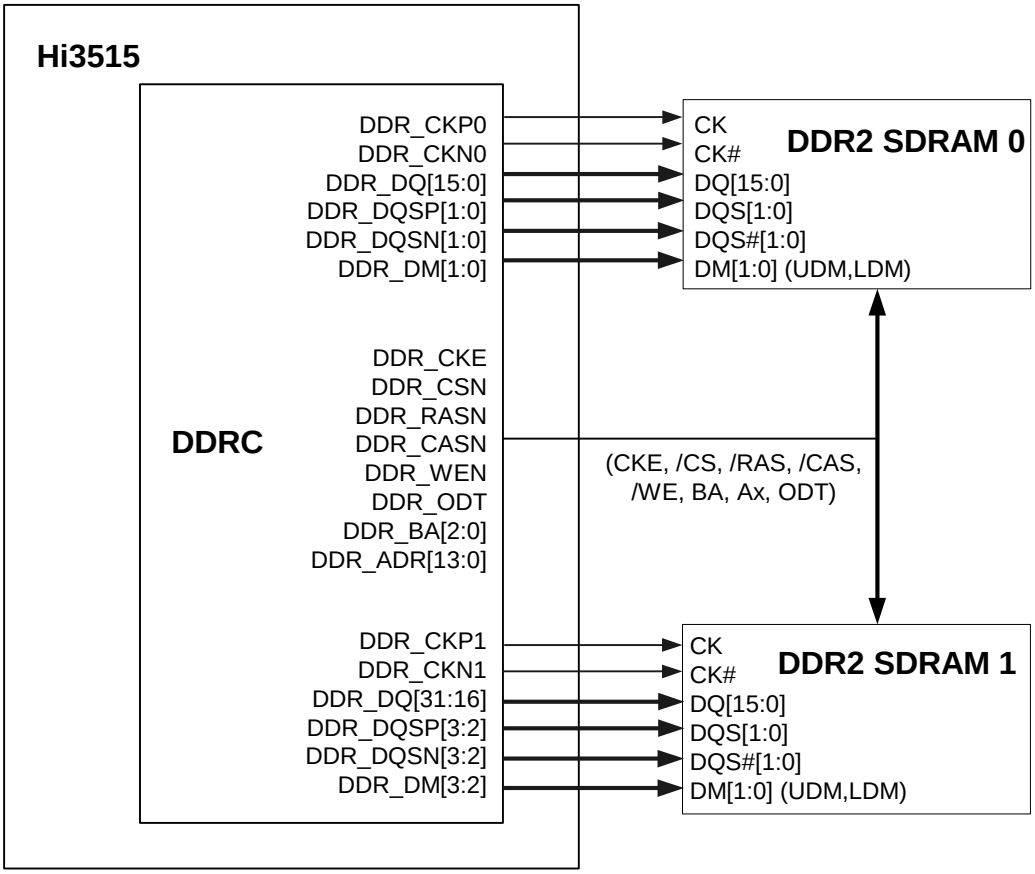
4.1.4 Function Description

4.1.4.1 Application Block Diagram

Through the DDRC, the master devices such as the CPU of the system-on-chip (SoC) can access the external DDR2 SDRAM. After the timing parameter registers of the DDRC are configured through the CPU, the DDRC supports the DDR2 SDRAM that complies with the JEDEC (JESD79) standard.

The DDRC can interconnect to the DDR2 SDRAM in 32-bit mode. In this mode, the DDRC connects to two DDR2 SDRAMs (each one has 16-bit data bus width), as shown in Figure 4-1.

Figure 4-1 Interconnection between the DDRC and two DDR2 SDRAMs



NOTE

- The symbol # indicates active low.
- Both DDR2 SDRAM0 and DDR2 SDRAM1 have 16-bit data bus width. Figure 4-1 shows how the DDRC connects to DDR2 SDRAMs in 32-bit DDR2 mode.
- The DDRC signals DDRCKE, DDRCSN, DDRRASN, DDRCASN, DDRWEN, DDRBA[2:0], and DDRADR[13:0] are connected to the command control signals CKE, /CS, /RAS, /CAS, /WE, BA, and Ax of DDR2 SDRAM0 respectively. See Figure 4-1.
- When the capacity of the DDR2 SDRAM is less than 1 Gbit, the signal DDR_BA[2] of the DDRC is floated as output.

4.1.4.2 Function Principle

The DDRC supports the DDR2 SDRAMs provided by the mainstream DRAM vendors, as shown in Table 4-2. The descriptions in Table 4-2 are based on the working frequencies of DDR2 SDRAMs. The restrictions such as the capacity are not taken into account.

Table 4-2 DDR2 SDRAMs supported by the DDRC

Vendor	200 MHz	333 MHz	400 MHz	Remarks
JESD79 (DDR2 Standard)	DDR2-400 DDR2-533 DDR2-667 DDR2-800	DDR2-667 DDR2-800	DDR2-800	1, 2
Micron	-5E DDR2-400 -37E DDR2-533 -3 DDR2-667 -3E DDR2-667 -25 DDR2-800 -25E DDR2-800	-3 DDR2-667 -3E DDR2-667 -25 DDR2-800 -25E DDR2-800	-25 DDR2-800 -25E DDR2-800	3
ELPIDA	-4A DDR2-400 -5C DDR2-533 -6E DDR2-667 -6C DDR2-667 -8E DDR2-800	-6E DDR2-667 -6C DDR2-667 -8E DDR2-800	-8E DDR2-800	3
Hynix	-E3 DDR2-400 -C4 DDR2-533 -Y4 DDR2-667 -Y5 DDR2-667 -S5 DDR2-800 -S6 DDR2-800	-Y4 DDR2-667 -Y5 DDR2-667 -S5 DDR2-800 -S6 DDR2-800	-S5 DDR2-800 -S6 DDR2-800	3
Samsung	-CC DDR2-400 -D5 DDR2-533 -E6 DDR2-667 -E7 DDR2-800	-E6 DDR2-667 -E7 DDR2-800	-E7 DDR2-800	3

NOTE

- 1. The Hi3515 DDRC supports most SDRAMs that comply with the JESD79 standard. In different operating modes, the DDRC supports only the SDRAMs whose frequencies are equal to or higher than the working frequency of the DDRC. When you use the SDRAMs provided by other vendors, you can select the SDRAMs and their classes based on the preceding descriptions.
- 2. The DDR2 SDRAMs provided by each vendor vary according to frequencies. Even at the same working frequency, the SDRAMs provided by the same vendor have different capacities and versions. The DDRC supports all the DDR2 SDRAM types listed in Table 4-2. You can select the capacity and bit width of a DDR2 SDRAM based on the application scenario of the system chip.
- 3. The DDRC supports the burst-of-four DDR2 SDRAMs only.

The functions of the DDRC are implemented according to the timings of the JESD79 DDR2 SDRAM. By transmitting the command words of the DDR2 SDRAM, the DDRC accesses the

DDR2 SDRAM and controls its status. Such functions include write/read access, auto refresh, and low-power control for the DDR2 SDRAM.

Command Truth Table

The DDRC supports the major command words of the DDR2 SDRAM. Table 4-3 shows the command truth table of the DDRC.

Table 4-3 Command truth table of the DDRC

FUNCTION	DDR CKE	DDR CSN	DDR RASN	DDR CASN	DDR WEN	DDRADR			DDRBA
						11	AP(10)	9:0	
DESELECT	H	H	X	X	X	X	X	X	X
NOP	H	L	H	H	H	X	X	X	X
ACTIVE	H	L	L	H	H	V	V	V	V
READ	H	L	H	L	H	V	V	V	V
WRITE	H	L	H	L	L	V	V	V	V
PRECHARGE	H	L	L	H	L	X	L	X	V
PRECHARGE ALL	H	L	L	H	L	X	H	X	X
AUTO Refresh	H	L	L	L	H	X	X	X	X
SELF Refresh	L	L	L	L	H	X	X	X	X
MODE REGISTER SET	H	L	L	L	L	V	V	V	V

NOTE

- 1 The AP bit in DDRADR column is used to check whether the current PRECHARGE command is for a single bank or for all banks. For the definition of the AP bit, see the DDR JEDEC standard.
- 2 H indicates high level; L indicates low level; V indicates valid; X indicates ignored.
- 3 The DDRC supports only the burst-of-four operation mode of the DDR2. It does not support the burst termination operation of the DDR2.

Auto-Fresh

When DDRC_TIMING2[taref] is set to a non-zero value, the DDRC refreshes the DDR2 SDRAM by generating a periodic auto refresh command automatically At normal temperature, the DDR2 SDRAM must be auto-refreshed for 8,192 times within 64 ms, that is, the auto-refresh cycle is 7.8 μs. The relationship between the configured value (Taref) of DDRC_TIMING2[taref] and the auto-refresh cycle T (T = 7.8 μs) is as follows:

$T \geq Taref \times 16$ DDR clock cycles

When DDRC_TIMING2[taref] is configured, the internal counter of the DDRC loads the taref value automatically and then counts in deremental mode. When the count value reaches 0, the DDRC initiates an auto-refresh operation and the counter reloads the taref value to count.

Low-Power Management

The DDRC supports the following two low-power management modes:

- Common low-power mode

  After the system is idle for pd_prd clock cycles, the DDRC forces the DDR2 SDRAM to enter the low-power mode.

  If DDRC_CONFIG[pd_en] and DDRC_CONFIG[pr_prd] are set to valid values, the DDRC also automatically forces the DDR2 SDRAM to enter the low-power mode if the system is idle. When DDRC_CONFIG[pd_en] is set to 1, the DDR2 SDRAM enters the low-power mode if the DDRC does not perform any access operation within DDRC_CONFIG[pr_prd] bus clock cycles.

- Self-refresh low-power mode

  To switch to the standby mode, you can force the DDR2 SDRAM to enter the self-refresh low-power mode by configuring the DDRC registers. In self-refresh low-power mode, the power consumption of the DDR2 SDRAM is reduced to the minimum level, but the data in the DDR2 SDRAM is retained. In this case, the system cannot access the DDR2 SDRAM.

  By configuring DDRC_CTRL[sr_req], the DDRC can force the DDR2 SDRAM to enter the low-power mode. To be specific, when DDRC_CTRL[sr_req] is set to 1, the DDRC forces the DDR2 SDRAM to enter the self-refresh mode and does not respond to the bus request after the DDRC completes the current access operation. When the DDRC_CTRL[sr_req] is set to 0, the DDR2 SDRAM exits the self-refresh mode.

Arbitration Mechanism

The DDRC uses the priority scheduling algorithm. That is, the DDRC adds priority attributes to bus commands by configuring DDRC_QOS[pri], and then schedules the commands based on the priority attributes. As a result, the DDR2 SDRAM is accessed in a high-effective manner. The DDRC can also add delay attributes to bus commands by configuring DDRC_QOS[qos_en] and DDRC_QOS[qos], thus ensuring delay response to the commands based on the delay priority scheduling algorithm.

Address Mapping Mode

The DDRC maps the system bus address to the address of the DDR2 SDRAM, so the system can access the DDR2 SDRAM. By configuring DDRC_CONFIG[mem_map], DDRC_CONFIG[mem_row], and DDRC_CONFIG[mem_col], the DDRC converts the system bus into the address of the DDR2 SDRAM based on the address mapping algorithm.

The following example describes the mapping algorithm of the system bus address and the address of the DDR2 SDRAM. Assume that the lower 29 bits of the system bus address are BUS_ADR[28:0], the valid address is BUS_ADR[m - 1:0], and the DDRC address of the Hi3515 is DDR_ADR[13:0]. Then, for the 1-Gbit DDR2 SDRAM, its row address is DDR_ROW[x - 1:0], column address of the DDR2 SDRAM is DDR_COL[y - 1:0], and bank address is DDR_BA[z - 1:0]. If the values of the row address, column address, and bank address are 13, 10, and 3 respectively, the values of x, y, and z are 13, 10, and 3 respectively. The width of the DDRC data storage bus is DW. In this case, the address mapping is as follows:

- When DDRC_CONFIG[mem_map] is set to 0, the row-bank-column (RBC) mapping mode is as follows:

  BUS_ADR[m - 1:0] = {DDR_ROW[x–1:0], DDR_BA[z - 1:0], DDR_COL[y - 1:0], DW{b0}}

- When DDRC_CONFIG[mem_map] is set to 1, the bank-row-column (BRC) mapping mode is as follows:

  BUS_ADR[m - 1:0]={DDR_BA[z - 1:0], DDR_ROW[x - 1:0], DDR_COL[y - 1:0], DW{b0}}

NOTE

In the preceding expressions, the condition of the equation m = x + y + z + DW is true.

When the DDRC is in 32-bit mode, the value of DW is 2.

In RBC mode, A10 acts as the AP function bit of the DDR. Table 4-4 shows the mapping between the system bus address and the address of the DDR2 SDRAM. In BRC mode, the addresses are mapped based on the preceding expressions.

Table 4-4 Address mapping when the DDRC is in 32-bit mode

Memory Type	Width of the	Width of the	DDRBA			Row Address or Column Address	DDRADR						
Mbit x bw	Row Address	Column Width	2	1	0		13	12	11	10/AP	9	8	[7:0]
256 Mbit 4 bank													
16 x 16	13	9	–	12	11	Row address	–	25	24	23	22	21	[20:13]
						Column address	–	–	–	AP	–	10	[9:2]
512 Mbit 4 bank													
32 x 16	13	10	–	13	12	Row address	–	26	25	24	23	22	[21:14]
						Column address	–	–	–	AP	11	10	[9:2]
1 Gbit 8 bank													
64 x 16	13	10	14	13	12	Row address	–	27	26	25	24	23	[22:15]
						Column address	–	–	–	AP	11	10	[9:2]
2 Gbit 8 bank													
128 x 16	14	10	14	13	12	Row address	28	27	26	25	24	23	[22:15]
						Column address	–	–	–	AP	11	10	[9:2]

4.1.5 Operating Mode

4.1.5.1 Clock Gating

After the system enters the low-power mode, the working clock of the DDRC can be disabled. Before the system restores to the normal mode, the working clock of the DDRC must be enabled.

To disable the DDRC clock, the following steps take place:

Step 1 Ensure that the system runs in the flash memory.

Step 2 Set DDRC_CONFIG[sr_cc] to 1 to enable the component clock control function.

Step 3 Set DDRC_CTRL[sr_req] to 1 to request to enter the self-refresh mode.

Step 4 Query the DDRC_STATUS[in_sr] bit until it is 1, and then go to Step 5.

Step 5 Disable the DDRC clock.

----End

To enable the DDRC clock, the following steps take place:

Step 1 Enable the DDRC clock when the system runs in normal mode.

Step 2 Wait until the internal DLL (delay lock loop) of the DDRC is locked.

Step 3 Set DDRC_CTRL[sr_req] to 0 to request to exit the self-refresh mode.

Step 4 Query the DDRC_STATUS[in_sr] bit until it is 0, and then go to Step 5.

Step 5 The DDRC clock works in normal mode.

----End

4.1.5.2 Soft Reset

The DDRC cannot be reset independently. It can be reset only when global soft reset is performed. After reset, the DDRC initializes the DDR2 SDRAM by following the initialization process.

4.1.5.3 Initialization

After power on, the system can access the DDR2 SDRAM only when the DDR2 SDRAM is initialized. Before initializing the DDR2 SDRAM, note the following points:

- Power on the DDR2 SDRAM according to the JEDEC standard. In other words, you must power on VDD, VDDQ, VREF, and VTT in sequence.
- Initialize the DDR2 SDRAM after the system runs in normal mode.

To initialize the DDR2 SDRAM, the following steps take place:

Step 1 The software waits for more than 200 μs.

Step 2 Set DDRC_CTRL to 0x0000_0004 to exit the self-refresh mode.

Step 3 The software waits for more than 400 ns.

Step 4 Set DDRC_EMRS01 to 0x0044_0662, set DDRC_EMRS23 to 0x0000_0000, and then configure the mode register and extended mode register of the DDR2 SDRAM. In this step, set the CAS latency (CL) of the DDR to 4 and set the burst length to 4. The value of the extended mode register can be set as required or set to the same value as the mode register of the DDR2 SDRAM. In this step, the value of the extended mode register is set to 0.

NOTE

- The values of the CL and DDRC_TIMING1[tcl] must be the same.
- DDRC_EMRS01 maps to the mode register (MRS) and extended mode register 1 (EMRS1) of the DDR2 SDRAM. You need to configure DDRC_EMRS01 according to the description of A15–A0 of the mode register in the DDR2 SDRAM manual. The three most significant bits (MSBs) of the mode register, namely, the bank address, are ignored.
- When configuring the extended mode register 1 (EMRS1) of the DDR2, the RDQS of RMRS1 must be disabled and $\overline{DQS}$ of EMRS1 must be enabled.

Step 5 Set DDRC_CONFIG to 0x7000_7022 based on the widths the row address and column address of the DDR2 SDRAM. This indicates that the address mapping mode is RBC, AP is A10, width of the column address is 10, width of the row address is 13, and seven auto-refresh operations are performed during initialization. In addition, DDRC_CONFIG[pd_en] is set to the power-on reset value, that is, the reset value is used in low-power mode.

NOTE

The low-power mode is disabled by default. The function of entering the low-power mode automatically must be disabled during initialization. In normal mode, you are recommended to enable the low-power control function, thus reducing power consumption.

Step 6 Configure DDRC_TIMING0, DDRC_TIMING1, DDRC_TIMING2, and DDRC_TIMING3. Note that the value of tcl and DDRC_EMRS01 must be the same.

Step 7 Configure DDRC_ODT_CONFIG. The default value is recommended. That is, odt is valid when the DDR2 is written; odt is invalid when the DDR2 is read.

Step 8 Configure DDRC_QOS of different ports based on the system requirements. It is recommended that the value of DDRC_QOS[qos] is greater than 1 and ranges from 0x002 to 0x3FF.

Step 9 Set DDRC_CTRL to 0x6 to start the initialization.

Step 10 After the value of DDRC_STATUS bit[3] changes to 1, it indicates that the initialization is complete.

----End

After the preceding steps are complete, the DDR2 SDRAM can work properly.

4.1.6 Registers Summary

Table 4-5 lists the DDRC registers.

Table 4-5 Summary of the DDRC register (base address: 0x2011_0000)

Offset Address	Register	Description	Page
0x00	DDRC_STATUS	DDRC status register	4-12
0x04	DDRC_CTRL	DDRC control register	4-13

Offset Address	Register	Description	Page
0x08	DDRC_EMRS01	DDR mode configuration register 1	4-14
0x0C	DDRC_EMRS23	DDR mode configuration register 2	4-15
0x10	DDRC_CONFIG	DDR configuration register	4-16
0x20	DDRC_TIMING0	DDRC timing register 0	4-18
0x24	DDRC_TIMING1	DDRC timing register 1	4-19
0x28	DDRC_TIMING2	DDRC timing register 2	4-20
0x2C	DDRC_TIMING3	DDRC timing register 3	4-22
0x3C	DDRC_DTR_CTRL	DDRC data training control register	4-22
0x40	DDRC_DTR_PRD	DDRC data training period register	4-23
0x44	DDRC_DTR_GATE	DDRC data training gating register	4-24
0x48	DDRC_DTR_LAT	DDRC data training delay register	4-26
0x4C	DDRC_PHY_STATUS	DDRC PHY status register	4-28
0x50	DDRC_PHY_ADDR	DDRC PHY training address register	4-28
0x54	DDRC_PHY_DATA0	DDRC PHY training data register 0	4-29
0x58	DDRC_PHY_DATA1	DDRC PHY training data register 1	4-29
0x5C	DDRC_PHY_DATA2	DDRC PHY training data register 2	4-30
0x60	DDRC_PHY_DATA3	DDRC PHY training data register 3	4-30
0x90	DDRC_IF_STATUS	DDRC interface status register	4-31
0x94	DDRC_ODT_CONFIG	DDRC ODT configuration register	4-31
0x100	DDRC_QOS_CTRL	DDRC quality of service (QoS) control register	4-32
0x104	DDRC_QOS_CONFIG	DDRC QoS configuration register	4-32
0x108+nx4	DDRC_QOS	DDRC QoS register	4-33
0x270	DDRC_PHY_CONFIG	DDRC PHY configuration register	4-34
0x274	DDRC_DLL_STATUS	DDRC_DLL status register	4-34

Offset Address	Register	Description	Page
0x278	DDRC_DLL_CONFIG	DDRC_DLL configuration register	4-35
0x27C	DDRC_IO_CTRL	DDRC IO configuration register	4-36

Table 4-6 lists the value ranges and meanings of the variables in the offset addresses of registers.

Table 4-6 Variables in the offset addresses of registers

Variable	Value Range	Description
n	0–15	QoS variable of the 16 IDs

4.1.7 Registers Description

DDRC_STATUS

DDRC_STATUS is the DDRC status register.

Offset Address	Register Name	Total Reset Value
0x00	DDRC_STATUS	0x0000_004D

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									test_pass	ecc_serr	ecc_derr	in_init	in_sr	reserved	busy
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1

Bits	Access	Name	Description
[31:7]	RO	reserved	Reserved.
[6]	RO	test_pass	Self-test status. 0: Do not pass the test. 1: Pass the test.
[5]	RC	ecc_serr	Single-bit error indicator through ECC. 0: No single-bit error occurs. 1: A single-bit error occurs.
[4]	RC	ecc_derr	Multi-bit error indicator through ECC. 0: No multi-bit error occurs. 1: A multi-bit error occurs.

[3]	RO	in_init	Initialization status indicator. 0: Being initialized. 1: Normal operating mode.
[2]	RO	in_sr	Self-refresh status indicator. 0: Normal. 1: Self refresh.
[1]	RO	reserved	Reserved.
[0]	RO	busy	Controller status. 0: Idle. 1: Busy.

DDRC_CTRL

DDRC_CTRL is the DDRC control register.

Offset Address	Register Name	Total Reset Value
0x04	DDRC_CTRL	0x0000_0001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											sref_test_req	ecc_en	clk_ratio	init_req	sr_req
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	Access	Name	Description
[31:5]	RO	reserved	Reserved.
[4]	RW	sref_test_req	Read/write data channel test of the controller and PHY. 0: Normal operating mode. 1: The self-test starts. When the test is complete, this bit is cleared automatically.
[3]	RW	ecc_en	CCC enable. Not supported.
[2]	RW	clk_ratio	Operating mode of the controller clock. 0: The controller and PHY work in 1:2 mode. 1: The controller and PHY work in 1:1 mode. Only the 1:1 mode is supported.

[1]	RW	init_req	Hardware initialization control. Only writing 1 to this bit has effect. 0: Normal operating mode. 1: The initialization starts. When the initialization is complete, this bit is cleared automatically.
[0]	RW	sr_req	Self-refresh request control. 0: Normal operating mode or request to exit the self-refresh mode when the DDRC is in self-refresh mode. 1: Request to enter the self-refresh mode.

DDRC_EMRS01

DDRC_EMRS01 is DDRC mode configuration register 1.

Offset Address	Register Name	Total Reset Value
0x08	DDRC_EMRS01	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved									ddr_rtt1	ddr_al		ddr_rtt0		ddr_drv	ddr_dll_en	reserved				ddr_wr			ddr_rst	ddr_tm		ddr_cas		ddr_bt		ddr_bl	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:23]	RW	reserved	Reserved, fixed at 0.
[22]	RW	ddr_rtt1	DDR2 SDRAM matched impedance. This bit works together with bit[ddr_rtt0].
[21:19]	RW	ddr_al	Reserved, fixed at 0.
[18]	RW	ddr_rtt0	DDR2 SDRAM matched impedance. This bit works together with bit[ddr_rtt0]. {ddr_rtt1,ddr_rtt0}: 00: Forbidden 01: 75 Ω 10: 150 Ω 11: 50 Ω Configure this bit according to the instructions in the DDR2 SDRAM manuals.
[17]	RW	ddr_drv	DDR2 SDRAM drive capability. 0: strong drive 1: weak drive

[16]	RW	ddr_dll_en	DDR2 SDRAM DLL enable. 0: enabled 1: disabled
[15:12]	RW	reserved	Reserved, fixed at 0.
[11:9]	RW	ddr_wr	DDR2 SDRAM write recovery time. 010: 3 clock cycles 011: 4 clock cycles 100: 5 clock cycles 101: 6 clock cycles Others: reserved
[8]	RW	ddr_rst	DDR2 SDRAM DLL reset control. 0: not reset 1: reset
[7]	RW	ddr_tm	DDR2 SDRAM test mode. 0: normal mode 1: test mode
[6:4]	RW	ddr_cas	DDR2 SDRAM CAS configuration. 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles 110: 6 clock cycles Others: reserved The CL cannot be set to 3. You must set the CL to a value greater than 3.
[3]	RW	ddr_bt	DDR2 SDRAM burst type. 0: in sequence 1: reserved
[2:0]	RW	ddr_bl	DDR2 SDRAM burst length. 010: 4 011: 8 Others: reserved

DDRC_EMRS23

DDRC_EMRS23 is DDR mode configuration register 2.

Offset Address	Register Name	Total Reset Value
0x0C	DDRC_EMRS23	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	emrs3																emrs2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	RW	emrs3	Reserved, fixed at 0.
[15:0]	RW	emrs2	Reserved, fixed at 0.

DDRC_CONFIG

DDRC_CONFIG is the DDR configuration register.

Offset Address	Register Name	Total Reset Value
0x10	DDRC_CONFIG	0x0000_6022

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	init_arefnum				pr_prd								reserved	sr_cc	reserved	pd_en	mem_type				reserved		ap	mem_map	mem_bank		mem_row		reserved		mem_col	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0

Bits	Access	Name	Description
[31:28]	RW	init_arefnum	Count of auto refreshes required for the hardware initialization. 0000–0010: 2 0011–1111: n clock cycles. The letter n indicates the corresponding decimal value.
[27:20]	RW	pr_prd	Count of wait CLKs for automatic power down. When the controller does not receive any request for a period, it disables the CKE of the DDR2 SDRAM. 0x00: When pd_en is valid and the controller does not receive any request at the next clock cycle, the DDR2 SDRAM enters the low-power mode automatically. 0x01–0xn: When pd_en is valid and the controller does not receive any request within n clock cycles, the DDR2 SDRAM enters the low-power mode automatically. For example, 0x03 indicates that the DDR2 SDRAM enters the low-power mode automatically if it does not receive any request within three clock cycles.
[19]	RO	reserved	Reserved.

[18]	RW	sr_cc	Self-refresh clock enable. 0: disabled 1: enabled
[17]	RO	reserved	Reserved.
[16]	RW	pd_en	Auto power-down enable. This signal is used to enable or disable the CKE of the DDR2 SDRAM automatically. When no read or write request is received within PD_PRD clock cycles, the CKE input of the DDR2 SDRAM is disabled automatically. 0: Do not enter the power-down mode automatically. 1: Enter the power-down mode automatically.
[15:12]	RW	mem_type	External memory type. 0000: 16-bit mobile SDRAM 0001: 32-bit mobile SDRAM 0010: 16-bit mobile DDR 0011: 32-bit mobile DDR 0100: 16-bit DDR Ⅰ 0101: 32-bit DDR Ⅰ 0110: 16-bit DDR Ⅱ 0111: 32-bit DDR Ⅱ Others: reserved
[11:10]	RO	reserved	Reserved.
[9]	RW	ap	Precharge all indicator. 0: a10 1: a8 (this value is used for early memories and rarely used for current memories)
[8]	RW	mem_map	Translation mode of the memory address. 0: Row, Bank, Col, Dw 1: Bank, Row, Col, Dw
[7]	RW	mem_bank	Number of memory banks. 0: 4 banks 1: 8 banks
[6:4]	RW	mem_row	Bit width of the row address for a single SDRAM. 000: 11 001: 12 010: 13 011: 14 100: 15 101: 16 Others: reserved

[3]	RO	reserved	Reserved.
[2:0]	RW	mem_col	Bit width of the column address for a single SDRAM. 000: 8 001: 9 010: 10 011: 11 100: 12 Others: reserved

DDRC_TIMING0

DDRC_TIMING0 is DDRC timing register 0.

Offset Address	Register Name	Total Reset Value
0x20	DDRC_TIMING0	0x7FFF_3F1F

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved		tmrd		trrd				trp				trcd				reserved		trc						reserved				tras			
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	1	1	1

Bits	Access	Name	Description
[31]	RO	reserved	Reserved.
[30:28]	RW	tmrd	Count of wait cycles of the command for loading the mode register (LMR). 000–001: 1 clock cycle 011–111: n clock cycles. The letter n indicates the corresponding decimal value. For example, the value 010 indicates two clock cycles.
[27:24]	RW	trrd	Count of wait cycles from ACT bank A to ACT bank B. 0000–0001: 1 clock cycle 0010–1111: n clock cycles. The letter n indicates the corresponding decimal value. For example, the value 1111 indicates 15 clock cycles.
[23:20]	RW	trp	Count of wait cycles of the disable command (PRE period). 0000–0001: 1 clock cycle 0010–1111: n clock cycles. The letter n indicates the corresponding decimal value. 1111: Reserved. For example, the value 0111 indicates seven clock cycles.

[19:16]	RW	trcd	Count of wait cycles from the ACT bank command to the read or write bank command. 0000–0011: 3 clock cycles 0100–1111: n clock cycles. The letter n indicates the corresponding decimal value. For example, the value 0011 indicates three clock cycles.
[15:14]	RO	reserved	Reserved.
[13:8]	RW	trc	Count of wait cycles from an ACT bank command to the next ACT bank command. 0x00–0x01: 1 clock cycle 0x02–0x0F: n clock cycles. The letter n indicates the corresponding decimal value. For example, 0x0F indicates 15 clock cycles.
[7:5]	RO	reserved	Reserved.
[4:0]	RW	tras	Count of wait cycles from the ACT bank command to the PRE disable command. 0x00–0x01: 1 clock cycle 0x00–0x0F: n clock cycles. The letter n indicates the corresponding decimal value. For example, 0x0F indicates 15 clock cycles.

DDRC_TIMING1

DDRC_TIMING1 is DDRC timing register 1.

Offset Address	Register Name	Total Reset Value
0x24	DDRC_TIMING1	0xFF01_23FF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	tsre								trdlat				reserved	trtw			twl				reserved	tcl			trfc							
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1

Bits	Access	Name	Description
[31:24]	RW	tsre	Count of wait cycles from the self-refresh exit command to the read command. 0x00–0xFF: n clock cycles. The letter n indicates the corresponding decimal value. For example, 0xFF indicates 255 clock cycles. In general, the DDR2 SDRAM requires 200 clock cycles.

[23:20]	RW	trdlat	Read data delay. This parameter depends on the length of the DQS trace. 0000–1111: n+1 clock cycles For example, 0000 indicates the delay of one clock cycle.
[19]	RO	reserved	Reserved.
[18:16]	RW	trtw	Delay from the last obtained data segment to the first written data segment. 000–110: n+1 clock cycles
[15:12]	RW	twl	Count of wait cycles from the write command to first data write valid. 0x0–0xF: n clock cycles. The letter n indicates the corresponding decimal value. For example, 0x3 indicates three clock cycles. In DDR2 mode, twl is set to (tcl – 1). The value of twl must meet the following condition: twl - taond ≥ 1
[11]	RO	reserved	Reserved.
[10:8]	RW	tcl	DDR CL from the read command to read data start. 100: CL = 4 101: CL = 5 110: CL = 6 Others: reserved
[7:0]	RW	trfc	Count of wait cycles of the AREF period or AREF to the ACT command. 0x00–0x01: 1 clock cycle 0x02–0xFF: n clock cycles. The letter n indicates the corresponding decimal value. For example, 0x0F indicates 15 clock cycles.

DDRC_TIMING2

DDRC_TIMING2 is DDRC timing register 2.

Offset Address	Register Name	Total Reset Value
0x28	DDRC_TIMING2	0x33F3_F7FF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved		tcke		reserved		twtr		twr				reserved				tfaw				reserved				taref							
Reset	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1

Bits	Access	Name	Description
[31:30]	RO	reserved	Reserved.
[29:28]	RW	tcke	Minimum period of maintaining the DDR low-power mode. 00–01: 1 clock cycle 10–11: n clock cycles. The letter n indicates the corresponding decimal value.
[27:26]	RO	reserved	Reserved.
[25:24]	RW	twtr	Count of wait cycles of the last write-to-read command. 00-01: 1 clock cycle 10–11: n clock cycles. The letter n indicates the corresponding decimal value. For example, the value 11 indicates three clock cycles.
[23:20]	RW	twr	Count of wait cycles of write recovery. 0x0–0x1: 1 clock cycle 0x2–0xF: n clock cycles. The letter n indicates the corresponding decimal value. For example, 0x7 indicates seven clock cycles.
[19:18]	RO	reserved	Reserved.
[17:12]	RW	tfaw	Count of clock cycles of four continuous ACT commands. 0x00−0x3F: n clock cycles For example, 0x14 indicates 20 clock cycles.
[11]	RO	reserved	Reserved.
[10:0]	RW	taref	Auto-refresh cycle. 0x000: forbidden 0x001–0x7FF: The auto-refresh cycle of the SDRAM is 16 x n clock cycles. Note: The letter n indicates the corresponding decimal value. For example, 0x008 indicates 128 clock cycles. The value depends on the operating frequency of the SDRAM.

DDRC_TIMING3

DDRC_TIMING3 is DDRC timing register 3.

Offset Address	Register Name	Total Reset Value
0x2C	DDRC_TIMING3	0x0000_0F02

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

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21:20]	RW	taond	ODT enable/disable cycle (Taond/Taofd). 00: 2/2.5 01: 3/3.5 10: 4/4.5 11: 5/5.5
[19:12]	RO	reserved	Reserved.
[11:8]	RW	txard	Count of wait cycles of exiting the DDR2 SDRAM low-power mode. 0x0–0xF: n clock cycles. The letter n indicates the corresponding decimal value. For example, 0x7 indicates seven clock cycles. The maximum value between tXP, tXARD, and tXARDS is used.
[7:3]	RO	reserved	Reserved.
[2:0]	RW	trtp	Wait delay from the read command to the disable command. 000–010: 2 clock cycles 011–111: n clock cycles The delay of DDR2 is calculated as follows: AL + BL/2 + Max(trtp, 2) - 2

DDRC_DTR_CTRL

DDRC_DTR_CTRL is the DDRC data training control register.

Offset Address	Register Name	Total Reset Value
0x3C	DDRC_DTR_CTRL	0x0000_0001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dt_byte								reserved																		dt_limit		reserved	track_en	train_en	itm_rst	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	Access	Name	Description
[31:24]	RW	dt_byte	Byte training enable. 0: disabled 1: enabled The least significant bit (LSB) maps to byte 0 of the DDR SDRAM.
[23:6]	RO	reserved	Reserved.
[5:4]	RW	dt_limit	DQS gating shift control. 00: shift 0° 01: shift 90° 10: shift 180° 11: shift 270°
[3]	RO	reserved	Reserved.
[2]	RW	track_en	Auto refresh enable for the gating position. 0: disabled 1: enabled
[1]	RW	train_en	Gating position training enable. 0: disabled 1: enabled
[0]	RW	itm_rst	ITM reset signal of the PHY. 0: valid 1: invalid

DDRC_DTR_PRD

DDRC_DTR_PRD is the DDRC data training period register.

Offset Address	Register Name	Total Reset Value
0x40	DDRC_DTR_PRD	0x0000_0000

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

Bits	Access	Name	Description
[31:20]	RW	reserved	Reserved.
[19:0]	RW	track_prd	Auto refresh cycle of the DQS gating. If there is no read operation during the configured clock cycles, the controller automatically initiates a read operation to refresh the DQS gating. 0x0: The controller initiates a dummy read if there is no read operation in one auto-refresh clock cycle. 0x1: The controller initiates a dummy read if there is no read operation in two auto-refresh clock cycles. 0x2–0x7FF: The controller initiates a dummy read if there is no read operation in n+1 auto-refresh clock cycles.

DDRC_DTR_GATE

DDRC_DTR_GATE is the DDRC data training gating register.

Offset Address	Register Name	Total Reset Value
0x44	DDRC_DTR_GATE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																gate_sel7		gate_sel6		gate_sel5		gate_sel4		gate_sel3		gate_sel2		gate_sel1		gate_sel0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:14]	RW	gate_sel7	Gating delay phase of DDR SDRAM byte 7. 00: 0° 01: 90° 10: 180° 11: 270°

[13:12]	RW	gate_sel6	Gating delay phase of DDR SDRAM byte 6. 00: 0° 01: 90° 10: 180° 11: 270°
[11:10]	RW	gate_sel5	Gating delay phase of DDR SDRAM byte 5. 00: 0° 01: 90° 10: 180° 11: 270°
[9:8]	RW	gate_sel4	Gating delay phase of DDR SDRAM byte 4. 00: 0° 01: 90° 10: 180° 11: 270°
[7:6]	RW	gate_sel3	Gating delay phase of DDR SDRAM byte 3. 00: 0° 01: 90° 10: 180° 11: 270°
[5:4]	RW	gate_sel2	Gating delay phase of DDR SDRAM byte 2. 00: 0° 01: 90° 10: 180° 11: 270°
[3:2]	RW	gate_sel1	Gating delay phase of DDR SDRAM byte 1. 00: 0° 01: 90° 10: 180° 11: 270°
[1:0]	RW	gate_sel0	Gating delay phase of DDR SDRAM byte 0. 00: 0° 01: 90° 10: 180° 11: 270°

DDRC_DTR_LAT

DDRC_DTR_LAT is the DDRC data training delay register.

Offset Address	Register Name	Total Reset Value
0x48	DDRC_DTR_LAT	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	tphy_rden7			reserved	tphy_rden6			reserved	tphy_rden5			reserved	tphy_rden4			reserved	tphy_rden3			reserved	tphy_rden2			reserved	thpy_rden1			reserved	tphy_rden0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RO	reserved	Reserved.
[30:28]	RW	tphy_rden7	Gating delay cycle of DDR SDRAM byte 7. 000: 0 clock cycles 001: 1 clock cycle 010: 2 clock cycles 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles Others: reserved
[27]	RO	reserved	Reserved.
[26:24]	RW	tphy_rden6	Gating delay cycle of DDR SDRAM byte 6. 000: 0 clock cycles 001: 1 clock cycle 010: 2 clock cycles 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles Others: reserved
[23]	RO	reserved	Reserved.
[22:20]	RW	tphy_rden5	Gating delay cycle of DDR SDRAM byte 5. 000: 0 clock cycles 001: 1 clock cycle 010: 2 clock cycles 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles Others: reserved

[19]	RO	reserved	Reserved.
[18:16]	RW	tphy_rden4	Gating delay cycle of DDR SDRAM byte 4. 000: 0 clock cycles 001: 1 clock cycle 010: 2 clock cycles 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles Others: reserved
[15]	RO	reserved	Reserved.
[14:12]	RW	tphy_rden3	Gating delay cycle of DDR SDRAM byte 3. 000: 0 clock cycles 001: 1 clock cycle 010: 2 clock cycles 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles Others: reserved
[11]	RO	reserved	Reserved.
[10:8]	RW	tphy_rden2	Gating delay cycle of DDR SDRAM byte 2. 000: 0 clock cycles 001: 1 clock cycle 010: 2 clock cycles 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles Others: reserved
[7]	RO	reserved	Reserved.
[6:4]	RW	thpy_rden1	Gating delay cycle of DDR SDRAM byte 1. 000: 0 clock cycles 001: 1 clock cycle 010: 2 clock cycles 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles Others: reserved
[3]	RO	reserved	Reserved.

[2:0]	RW	tphy_rden0	Gating delay cycle of DDR SDRAM byte 0. 000: 0 clock cycles 001: 1 clock cycle 010: 2 clock cycles 011: 3 clock cycles 100: 4 clock cycles 101: 5 clock cycles Others: reserved

DDRC_PHY_STATUS

DDRC_PHY_STATUS is the DDRC PHY status register.

Offset Address	Register Name	Total Reset Value
0x4C	DDRC_PHY_STATUS	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	dtr_status	Phase shift status of the byte gating position. 00: shift 0° 01: shift 90° 10: shift 180° 11: 270°
[15:8]	RO	dtr_err	Auto tracking error status of the gating position. 0: No error occurs during tracking. 1: An error occurs during tracking.
[7:0]	RO	dtr_ok	Gating training status. 0: An error occurs during training. 1: No error occurs during training.

DDRC_PHY_ADDR

DDRC_PHY_ADDR is the DDRC PHY training address register.

Offset Address	Register Name	Total Reset Value
0x50	DDRC_PHY_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RW	train_row	Gating training position, the row address of the DDR2 SDRAM is used. When the row address is less than 16 bits, the upper bits are stuffed with 0s.
[15:13]	RW	train_bank	Gating training position, the bank address of the DDR2 SDRAM is used. When the bank address is less than three bits, the upper bits are stuffed with 0s.
[12:0]	RW	train_col	Gating training position, the column address of the DDR2 SDRAM is used. When the column address is less than 13 bits, the upper bits are stuffed with 0s.

DDRC_PHY_DATA0

DDRC_PHY_DATA0 is DDRC PHY training data register 0.

Offset Address	Register Name	Total Reset Value
0x54	DDRC_PHY_DATA0	0xEE11_DD22

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dt_data0																															
Reset	1	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	1	1	0	1	1	1	0	1	0	0	1	0	0	0	1	0

Bits	Access	Name	Description
[31:0]	RW	dt_data0	DQS gating training data 0.

DDRC_PHY_DATA1

DDRC_PHY_DATA1 is DDRC PHY training data register 1.

Offset Address	Register Name	Total Reset Value
0x58	DDRC_PHY_DATA1	0xCC33_BB44

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dt_data1																															
Reset	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	0	1	0	0	0	1	0	0

Bits	Access	Name	Description
[31:0]	RW	dt_data1	DQS gating training data 1.

DDRC_PHY_DATA2

DDRC_PHY_DATA2 is DDRC PHY training data register 2.

Offset Address	Register Name	Total Reset Value
0x5C	DDRC_PHY_DATA2	0x44BB_33CC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dt_data2																															
Reset	0	1	0	0	0	1	0	0	1	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0

Bits	Access	Name	Description
[31:0]	RW	dt_data2	DQS gating training data 2.

DDRC_PHY_DATA3

DDRC_PHY_DATA3 is DDRC PHY training data register 3.

Offset Address	Register Name	Total Reset Value
0x60	DDRC_PHY_DATA3	0x22DD_11EE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dt_data3																															
Reset	0	0	1	0	0	0	1	0	1	1	0	1	1	1	0	1	0	0	0	1	0	0	0	1	1	1	1	0	1	1	1	0

Bits	Access	Name	Description
[31:0]	RW	dt_data3	DQS gating training data 3.

DDRC_IF_STATUS

DDRC_IF_STATUS is the DDRC interface status register.

Offset Address	Register Name	Total Reset Value
0x90	DDRC_IF_STATUS	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1:0]	RO	axistatus	Status indicator of the DDRC bus interface. 0: Idle. 1: A command is being executed. Each bit indicates the status of a DDRC interface. For example, 0x01 indicates that there is data in the buffer of the ahb0 interface.

DDRC_ODT_CONFIG

DDRC_ODT_CONFIG is the DDRC ODT configuration register.

Offset Address	Register Name	Total Reset Value
0x94	DDRC_ODT_CONFIG	0x0000_0001

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1]	RW	rodt	Read ODT enable. 0: disabled 1: enabled
[0]	RW	wodt	Write ODT enable. 0: disabled 1: enabled

DDRC_QOS_CTRL

DDRC_QOS_CTRL is the DDRC QoS control register.

Offset Address	Register Name	Total Reset Value
0x100	DDRC_QOS_CTRL	0x0000_0888

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				adp_cnt												reserved				dmc_fifo_lvl				axi1_fifo_lvl				axi0_fifo_lvl			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Bits	Access	Name	Description
[31:27]	RO	reserved	Reserved.
[26:17]	RW	adp_cnt	Priority adaptation cycle. This register is used to configure the cycle of changing the priority. When each adp_cntT cycles after the command in the DMC times out, the absolute priority of this command raises by one level until the priority becomes the highest priority. 0x0: disabled 0x1–0x3FF: enabled
[16:12]	RO	reserved	Reserved.
[11:8]	RW	dmc_fifo_lvl	Depth of the command register FIFO in the DMC. 0x0-0x1: depth of one command 0x2-0x8: depths of n commands Others: reserved
[7:4]	RW	axi1_fifo_lvl	Full command threshold for AXI interface 1, used for controlling the traffic of interface commands. 0x0-0x1: depth of one command 0x2-0x8: depths of n commands Others: reserved
[3:0]	RW	axi0_fifo_lvl	Full command threshold for AXI interface 0, used for controlling the traffic of interface commands. 0x0-0x1: depth of one command 0x2-0x8: depths of n commands Others: reserved

DDRC_QOS_CONFIG

DDRC_QOS_CONFIG is the DDRC QoS configuration register.

Offset Address	Register Name	Total Reset Value
0x104	DDRC_QOS_CONFIG	0x0000_3210

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	RW	id_map	QoS that is configured by selecting four bits based on the bus ID. Bit[15:12]: configure bit[3] mapping to the ID Bit[11:8]: configure bit[2] mapping to the ID Bit[7:4]: configure bit[1] mapping to the ID Bit[3:0]: configure bit[0] mapping to the ID For example, if ID_MAP is set to 0x5320, it indicates that ID[5], ID[3], ID[2], and ID[0] of the bus ID are used for ID mapping and QoS configuration.

DDRC_QOS

DDRC_QOS is the DDRC QoS register.

Offset Address	Register Name	Total Reset Value
0x108+nx4 (n = 0–15)	DDRC_QOS	0x0000_0000

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

Bits	Access	Name	Description
[31:17]	RO	reserved	Reserved.
[16]	RW	qos_en	QoS enable. 0: disabled 1: enabled
[15:14]	RO	reserved	Reserved.
[13:4]	RW	qos	Channel aging time. The recommended value ranges from 0x2 to 0x3FF.
[3]	RO	reserved	Reserved.

[2:0]	RW	pri	Channel priority. 000: highest priority 001: higher priority … 111: lowest priority Priority sequence: 000 > 001 > … > 111

DDRC_PHY_CONFIG

DDRC_PHY_CONFIG is the DDRC_PHY configuration register.

Offset Address	Register Name	Total Reset Value
0x270	DDRC_PHY_CONFIG	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																												ioodt		ck1_dis	ck0_dis
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:4]	RW	reserved	Reserved.
[3:2]	RW	ioodt	ODT configuration in the DDR2 IO. 00: Disable the ODT function of the IO 01: 75 Ω 10: 150 Ω 11: reserved
[1]	RW	ck1_dis	DDR CK1 low-power control. 0: Do not disable the clock output of CK1 1: Disable the clock output of CK1
[0]	RW	ck0_dis	DDR CK0 low-power control. 0: Do not disable the clock output of CK0 1: Disable the clock output of CK0

DDRC_DLL_STATUS

DDRC_DLL_STATUS is the DDRC_DLL status register.

Offset Address	Register Name	Total Reset Value
0x274	DDRC_DLL_STATUS	0x0000_00F0

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

Bits	Access	Name	Description
[31:12]	RO	reserved	Reserved.
[11:4]	RO	dll_mdly_tap	Valid delay unit level of the master DLL. This register indicates the DLL delay units required for delaying a clock cycle.
[3:2]	RO	reserved	Reserved.
[1]	RO	overflow	DLL overflow indicator. When the total delay of all the delay units is less than a clock cycle, the DLL sends an overflow indicator. 0: DLL overflow does not occur. 1: DLL overflow occurs.
[0]	RO	locked	DLL lock indicator. 0: Unlocked. 1: Locked.

DDRC_DLL_CONFIG

DDRC_DLL_CONFIG is the DDRC_DLL configuration register.

Offset Address	Register Name	Total Reset Value
0x278	DDRC_DLL_CONFIG	0x0000_0009

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																dll_slave_sel								reserved	dll_mode	cali_en	stop	tune			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:8]	RW	dll_slave_sel	Delay level of the slave DLL. The maximum value is 2D. The value is valid when DLL_MODE is 1.
[7]	RO	reserved	Reserved.

[6]	RW	dll_mode	Operating mode of the slave DLL. 0: The delay level of the slave DLL is controlled by the master DLL automatically. 1: The delay level of the slave DLL is equal to the value of the DLL_SDLY_SEL register.
[5]	RW	cali_en	DLL re-calibration enable (for frequency switching). 0: disabled 1: enabled
[4]	RW	stop	DLL stop tracking control. 0: The DLL tracks dynamically. 1: The DLL stops tracking.
[3:0]	RW	tune	DLL delay tune. Bit[3] 0: Increase the DLL delay value. 1: Decrease the DLL delay value. Bit[2:0] 000: 0 delay units 001: 1 delay unit … 111: 7 delay units By configuring this register, you can increase or decrease the DLL delay.

DDRC_IO_CTRL

DDRC_IO_CTRL is the DDRC_IO_CTRL configuration register.

Offset Address	Register Name	Total Reset Value
0x27C	DDRC_IO_CTRL	0x0001_0001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved														pwd_inout	pwd_out	reserved			lvcmos	reserved		drv_ck		reserved		drv_cmd		reserved		drv_data	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	Access	Name	Description
[31:18]	RO	reserved	Reserved.
[17]	RW	pwd_inout	PWD control for the DDR IO (including DQ and DQS). 0: Power down is invalid. 1: Power down is valid.

[16]	RW	pwd_out	PWD control for the DDR output (including CK, CMD, ADDR, and DM). 0: Power down is invalid. 1: Power down is valid.
[15:13]	RO	reserved	Reserved.
[12]	RW	lvcmos	0: SSTL18 level, DDRII mode. 1: 1.8 V mobile DDR mode. For the Hi3515, this bit must be set to 0. That is, the mobile DDR mode is not supported.
[11:10]	RO	reserved	Reserved.
[9:8]	RW	drv_ck	Drive capability of the CK pin (S0, S1). DDR2 mode 00: SSTL18 full strength 01: SSTL18 half strength Others: reserved
[7:6]	RO	reserved	Reserved.
[5:4]	RW	drv_cmd	Drive capability of the CMD and ADDR pins (S0, S1). DDR2 mode (LVCMOS = 0) 00: SSTL18 full strength 01: SSTL18 half strength Others: reserved
[3:2]	RO	reserved	Reserved.
[1:0]	RW	drv_data	Drive capability of the DQ, DM, and DQS pins (S0, S1). DDR2 mode (LVCMOS = 0) 00: SSTL18 full strength 01: SSTL18 half strength Others: reserved

4.2 SMI Controller

4.2.1 Overview

The static memory interface (SMI) controller provides asynchronous static memory interfaces, which are connected to asynchronous static memories to boot the system and store data. The asynchronous static memories include static random access memories (SRAMs), pseudo static random access memories (PSRAMs), read-only memories (ROMs), and NOR flash memories. The SMI controller can also be connected to the control chip with the asynchronous static memory interface to implement the main control function.

4.2.2 Features

The SMI controller has the following features:

- Supports asynchronous static memories, including SRAMs, PSRAMs, ROMs, and NOR flash memories.
- Supports asynchronous page read operation.
- Supports the configurable working clock.
- Supports two individual memory interfaces: bank 0 and bank 1.
- Supports the booting from bank 0 external memory.
- Each memory interface supports an external memory with 8-bit data width.
- Each memory interface supports the size of up to 256 Mbits when it is connected to an external memory with 8-bit data width.
- Supports configurable read wait cycle (TWSTRD), write wait cycle (TWSTWR), and subsequent burst read wait cycle (TWSTBURSTRD). Up to 31 SMI working clock cycles can be configured.
- Supports configurable read (output) enable wait cycle (TWSTOEN) and write enable wait cycle (TWSTWEN). Up to 15 SMI working clock cycles can be configured.
- Supports configurable bus turnaround (idle) cycle TWSTIDLY between read and write memory accesses. Up to 15 SMI working clock cycles can be configured.
- Supports the input of asynchronous wait signals and configuration of the valid polarity of these signals.
- Supports clock gating.

4.2.3 Signal Description

Table 4-7 Interface signals of the SMI controller

Signal	Direction	Description	Pin
SMI_CS0	O	Bank 0 chip select signal, active high or active low (the default is active low).	EBICS0N
SMI_CS1	O	Bank 1 chip select signal, active high or active low (the default is active low). It is multiplexed with the GPIO pin. For details about the multiplexing configuration information, see "Pin Multiplexing" in section 4.2.5 "Operating Mode."	EBICS1N
SMI_OEN	O	Read enable signal, active low.	EBIOEN
SMI_WEN	O	Write enable signal, active low.	EBIWEN

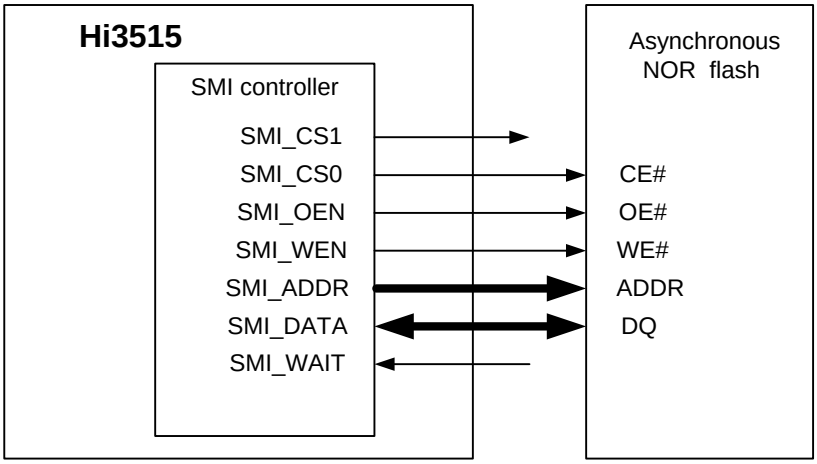
Signal	Direction	Description	Pin
SMI_ADDR[24:0]	O	Address bus. SMI_ADDR[24:15] is multiplexed with the BOOTSEL pin, FUNSEL pin, and NF pin. For details about the multiplexing configuration information, see "Pin Multiplexing" in section 4.2.5 "Operating Mode."	EBIADR24 to EBIADR0
SMI_DATA[7:0]	I/O	Bidirectional data bus.	EBIDQ7 to EBIDQ0
SMI_WAIT	I	External input synchronous wait signal, active high or active low (the default is active low). The pin is multiplexed with the IRRCV pin. For details about the multiplexing configuration information, see "Pin Multiplexing" in section 4.2.5 "Operating Mode."	EBIRDYN

4.2.4 Function Description

Application Block Diagram

The SMI controller controls data exchange between the internal system bus and the external asynchronous static memory bus. It provides flexible configuration of timing parameters used for connecting to various asynchronous static memories. See Figure 4-2.

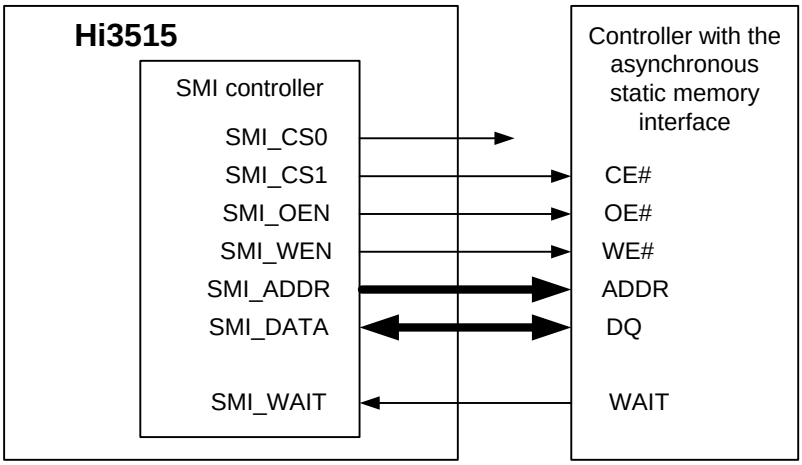
Figure 4-2 Connection between the SMI controller and the asynchronous static memory



Description:
The SMI controller outputs two individual chip select signals: SMI_CS0 and SMI_CS1. It also supports the booting from the external memory (8-bit data width) that is connected to the SMI_CS0.
When the SMI controller is connected to an asynchronous static memory, the SMI_WAIT signal is not used.

In addition, the SMI controller provides the asynchronous wait mechanism that is used to connect the control devices integrated with asynchronous static memory interfaces. See Figure 4-3.

Figure 4-3 Connection between the SMI controller and the control devices integrated with asynchronous static memory interfaces



Description:
The SMI controller outputs two individual chip select signals: SMI_CS0 and SMI_CS1. It also supports the booting from the external memory (8-bit data width) that is connected to the SMI_CS0.
When the SMI controller is connected to the control devices integrated with asynchronous static memory interfaces, the SMI_WAIT/WAIT signal is used for the handshake between them.

Function Principle

The SMI controller transforms the interface timing according to the timing parameters of the interconnected device. Through the change of the working clock of the SMI controller, the compatibility of the SMI controller is enhanced. In this way, the SMI controller can be compatible with as many static memories as possible. The configuration of timing parameters varies from working clock to working clock. See Table 4-8.

Table 4-8 Configuration of timing parameters of the SMI controller (bus clock fBUSCLK = 200 MHz)

Timing Parameter	f_{SMICLK} = 200 MHz	f_{SMICLK} = 100 MHz
Read wait cycle T_{WSTRD}(max)	155 ns	309 ns
Write wait cycle T_{WSTWR}(max)	155 ns	309 ns
Subsequent burst read wait cycle $T_{WSTBURSTRD}$(max)	155 ns	309 ns
Read (output) enable wait cycle T_{WSTOEN}(max)	75 ns	150 ns
Write enable wait cycle T_{WSTWEN}(max)	75 ns	150 ns
Bus turnaround (idle) cycle $T_{WSTIDLY}$(max) between read and write memory accesses	75 ns	150 ns

Note: The results in Table 4-8 are obtained after the time values are rounded towards minus infinity.

CAUTION

The SMI controller supports the booting from bank 0 only.

The asynchronous static memory interface is designed for universal use; therefore, the SMI controller provides the following two operating modes to meet the diversified requirements of interconnected devices:

- Timing parameter mode

  When the configuration of the timing configuration registers provided by the SMI controller meets the timing requirements of interconnected devices, the SMI controller works in this mode to transfer data to interconnection devices. By default, the SMI controller works in timing parameter mode.

- Asynchronous wait mode

  When the configuration of the timing configuration register provided by the SMI controller cannot meet the timing requirement of interconnected devices, or the timings of interconnected devices are not definite, the SMI controller works in this mode to transfer data to interconnected devices.

NOTE

When the SMI controller works in asynchronous wait mode, the interconnected devices must have corresponding asynchronous wait control signals.

Figure 4-4 shows the typical timing in timing parameter mode. It shows the timing of the SMI controller interface in a single read/write access.

Figure 4-4 Timing diagram (read/write) of the SMI controller in timing parameter mode

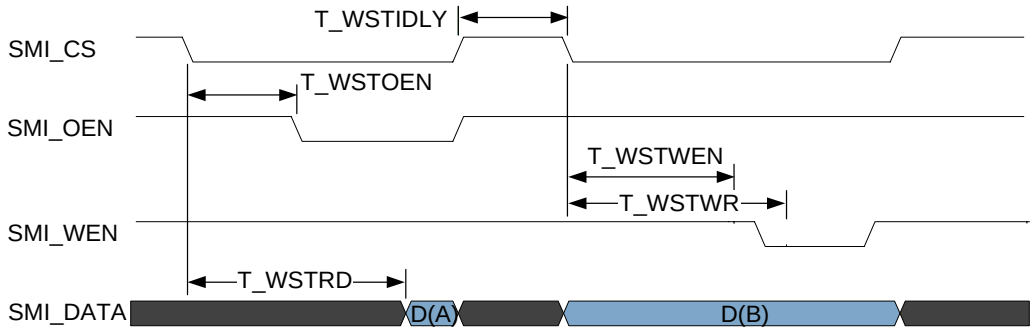


Table 4-9 lists the read/write timing parameters of the SMI controller.

Table 4-9 Read/write timing parameters of the SMI controller

Symbol	Min	Typ	Max	Description
T_WSTOEN	0	-	T_{WSTOEN}(max)	Chip select signal valid to read enable signal valid
T_WSTRD	-	-	T_{WSTRD}(max)	Chip select signal valid to read data valid
T_WSTIDLY	0	-	$T_{WSTIDLY}$(max)	Interval between two successive read/write operations
T_WSTWEN	0	-	T_{WSTWEN}(max)	Chip select signal valid to write enable signal valid
T_WSTWR	0	-	T_{WSTWR}(max)	Chip select signal valid to write data valid

Figure 4-5 shows the timing of the SMI controller interface in page read mode. The page read mode greatly improves the data read speed.

Figure 4-5 Timing diagram (page read) of the SMI controller in timing parameter mode

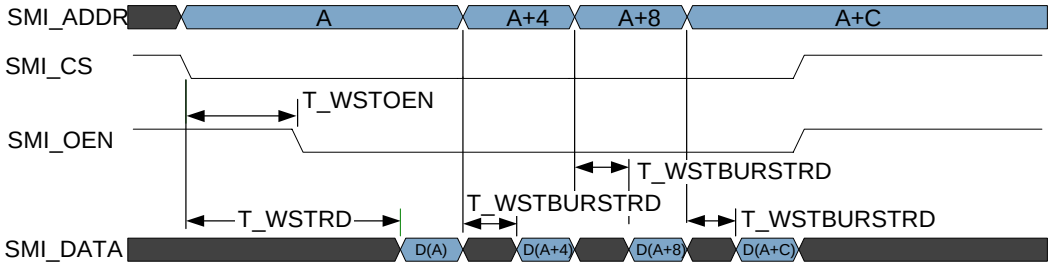


Table 4-10 lists the timing parameters of the SMI controller in page read mode.

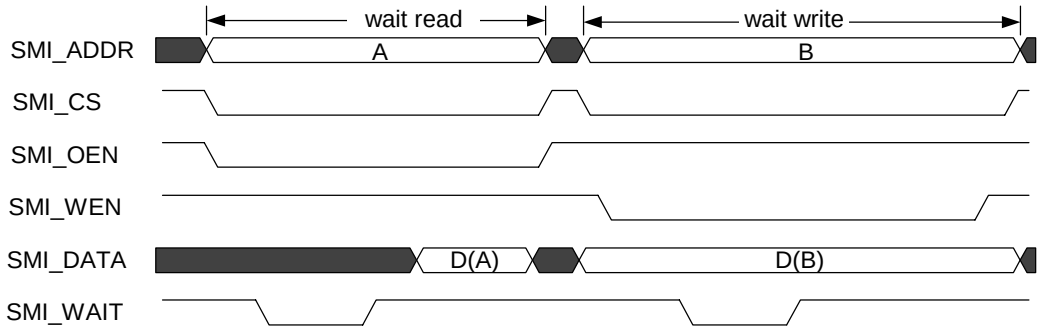
Table 4-10 Page read timing parameters of the SMI controller

Symbol	Min	Typ	Max	Description
T_WSTOEN	0	-	T_{WSTOEN}(max)	Chip select signal valid to read enable signal valid
T_WSTRD	-	-	T_{WSTRD}(max)	Chip select signal valid to read data valid
T_WSTBURSTRD	-	-	$T_{WSTBURSTRD}$(max)	Interval between two successive read operations in page read mode

Table 4-11 shows the typical timing in asynchronous wait mode.

Figure 4-6 shows the read/write timing when the SMI_WAIT signal is used for the handshake between the SMI controller and the external control chips.

Figure 4-6 Timing diagram (wait read/write) of the SMI controller in asynchronous wait mode



4.2.5 Operating Mode

Pin Multiplexing

The pins of the SMI controller are multiplexed with other pins. Therefore, before using SMI controller, the system controller must be configured to enable the SMI function of the corresponding pins. For details, see the configuration description of the I/O configuration registers reg42 and reg45.

Clock Gating

When the current data transfer is complete and a new data transfer does not start, the software can disable the clock of the SMI controller by writing 1 to SC_PERDIS[smiclkdis].

When the SMI controller is required for data transfer, the software can enable the clock of the SMI controller by the writing 1 to SC_PEREN[smiclken].

For details about how to query the current clock status of the SMI controller, see the descriptions of SC_PERCLKEN.

Clock Configuration

CAUTION

The clock ratio of the SMI_CR register must be configured same as that of SC_PERCTRL9[ssmcclk_sel].

When the current data transfer is complete and a new data transfer does not start, the software can configure the working clock of the SMI controller by controlling SC_PERCTRL9[ssmcclk_sel]. The frequency of the working clock of the SMI controller can be configured equal to that of the bus clock or equal to that of the bus clock divided by 2. By default, the frequency of the working clock of the SMI controller is equal to that of the bus clock.

Boot Configuration

The system can only boot from the external memory with 8-bit data width that is connected to SMI bank 0.

CAUTION

When the system boots from the external memory connected to the SMI bank 0, the chip select of the external memory must be active low.

Table 4-11 lists the memory address space supported by the SMI controller.

Table 4-11 Memory address space supported by the SMI controller

Bank ID	Address Space
Bank 0	0x8000_0000 to 0x83FF_FFFF
Bank 1	0x8400_0000 to 0x87FF_FFFF

Initialization Configuration of the SMI Controller in Timing Parameter Mode

The SMI controller needs to be configured properly according to the timing parameters and features of the external memory of the SMI controller. For details about the description of the timing parameters, see "Function " in 4.2.4 "Function Description."

To initialize the SMI controller in timing parameter mode (by taking bank 1 as an example), do as follows:

Step 1 Configure SMI_CR to set the working clock ratio of the SMI controller.

Step 2 Configure SMI_BIDCYR1[idcy] to set the bus turnaround (idle) cycle TWSTIDLY between read and write memory accesses.

Step 3 Configure SMI_BWSTRDR1[wstrd] to set the read wait cycle TWSTRD.

Step 4 Configure SMI_BWSTWRR1[wstwr] to set the write wait cycle TWSTWR.

Step 5 Configure SMI_BWSTOENR1[wstoen] to set the read (output) enable wait cycle TWSTOEN.

Step 6 Configure SMI_BWSTWENR1[wstwen] to set the write enable wait cycle TWSTWEN.

Step 7 Configure SMI_BWSTBRDR1[wstbrdr] to set the subsequent burst read wait cycle TWSTBURSTRD.

Step 8 Configure SMI_BCR1 to set memory bank control parameters.

---End

After the preceding steps are complete, the CPU can access the external memory through the configured SMI controller.

CAUTION

- When the SMI controller is connected to an external asynchronous static memory, the data transfer between the controller and the memory can be enabled only if the clock ratio SMI_CR[memclkratio] of the SMI register and the data width SMI_BCR1[mw] of the external memory are configured.
- When the software configures each of the memory banks, the following conditions must be met: $TWSTOE_N \leq TWSTRD$ and $TWSTWEN \leq TWSTWR$.

Initialization Configuration of the SMI Controller in Asynchronous Wait Mode

When the SMI controller is connected to a storage chip or control chip with wait signal output, the SMI controller can be switched to the asynchronous wait mode. In this case, data can be transferred through the handshake between the SMI controller and the external chip.

To initialize the SMI controller in asynchronous wait mode (by taking bank 1 as an example), do as follows:

Step 1 Configure the I/O configuration register (reg45) so that the EBIRDYN pin is used to input the SMI_WAIT signal.

Step 2 Configure the SMI_CR register to set the working clock ratio of the SMI controller.

Step 3 Configure SMI_BCR1 to set memory bank control parameters.

Step 4 Configure SMI_BCR1[waitpol] according to the valid polarity of the wait signal output by the external chip.

Step 5 Set SMI_BCR1[waiten] to 1 to enable the asynchronous wait mode of the SMI controller.

---End

After the preceding steps, the SMI controller can work in asynchronous wait mode.

4.2.6 Register Summary

Table 4-12 Summary of SMI controller registers (base address: 0x1010_0000)

Offset Address	Name	Description	Page
0x000	SMI_BIDCYR1	Idle cycle control register of SMI bank 1	4-47
0x004	SMI_BWSTRDR1	Read wait cycle control register of SMI bank 1	4-47
0x008	SMI_BWSTWRR1	Write wait cycle control register of SMI bank 1	4-48
0x00C	SMI_BWSTOENR1	Read enable assertion delay control register of SMI bank 1	4-49
0x010	SMI_BWSTWENR1	Write enable assertion delay control register of SMI bank 1	4-49
0x014	SMI_BCR1	Control register of SMI bank 1	4-50
0x018	SMI_BSR1	Status register of SMI bank 1	4-51
0x01C	SMI_BWSTBRDR1	Burst read wait delay control register of SMI bank 1	4-52
0x020–0x0DC	RESERVED	Reserved	-
0x0E0	SMI_BIDCYR0	Idle cycle control register of SMI bank 0	4-53
0x0E4	SMI_BWSTRDR0	Read wait cycle control register of SMI bank 0	4-53
0x0E8	SMI_BWSTWRR0	Write wait cycle control register of SMI bank 0	4-54
0x0EC	SMI_BWSTOENR0	Read enable assertion delay control register of SMI bank 0	4-55
0x0F0	SMI_BWSTWENR0	Write enable assertion delay control register of SMI bank 0	4-55
0x0F4	SMI_BCR0	Control register of SMI bank 0	4-56
0x0F8	SMI_BSR0	Status register of SMI bank 0	4-57
0x0FC	SMI_BWSTBRDR0	Burst read wait delay control register of SMI bank 0	4-58
0x100–0x1FC	RESERVED	Reserved	-
0x200	SMI_SR	SMI asynchronous wait status register	4-59
0x204	SMI_CR	SMI working clock configuration register	4-59

Offset Address	Name	Description	Page
0x208–0xFFC	RESERVED	Reserved	-

4.2.7 Register Description

SMI_BIDCYR1

SMI_BIDCYR1is the idle cycle control register of SMI bank 1. It controls the bus turnaround (idle) cycle between read and write memory accesses.

Offset Address	Register Name	Total Reset Value
0x000	SMI_BIDCYR1	0x0000_000F

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

Bits	Access	Name	Description
[31:4]	-	reserved	Reserved.
[3:0]	RW	idcy	$T_{WSTIDLY}$ controls the idle cycle between read and write. $T_{WSTIDLY} = idcy \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, idcy indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BWSTRDR1

SMI_BWSTRDR1 is the read wait cycle control register of SMI bank 1. It controls the read wait cycle.

Offset Address	Register Name	Total Reset Value
0x004	SMI_BWSTRDR1	0x0000_001F

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

Bits	Access	Name	Description
[31:5]	-	reserved	Reserved.
[4:0]	RW	wstrd	T_{WSTRD} controls the read wait cycle. For non-burst read, it controls the read wait cycle. For burst read, it controls the initial burst read wait cycle. The subsequent read wait cycles are configured through SMI_BWSTBRDR1. $T_{WSTRD} = wstrd \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstrd indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BWSTWRR1

SMI_BWSTWRR1 is the write wait cycle control register of SMI bank 1. It controls the write wait cycle.

Offset Address	Register Name	Total Reset Value
0x008	SMI_BWSTWRR1	0x0000_001F

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

Bits	Access	Name	Description
[31:5]	-	reserved	Reserved.
[4:0]	RW	wstwr	T_{WSTWR} controls the write wait cycle. $T_{WSTWR} = wstwr \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstwr indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BWSTOENR1

SMI_BWSTOENR1 is the read enable assertion delay control register of SMI bank 1. It controls the wait time from the output of the valid chip select signal to the output of the valid read enable signal.

Offset Address	Register Name	Total Reset Value
0x00C	SMI_BWSTOENR1	0x0000_000F

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

Bits	Access	Name	Description
[31:4]	-	reserved	Reserved.
[3:0]	RW	wstoen	T_{WSTOEN} controls the wait time from the output of the valid chip select signal to the output of the valid read enable signal. $T_{WSTOEN} = wstoen \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstoen indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BWSTWENR1

SMI_BWSTWENR1 is the write enable assertion delay control register of SMI bank 1. It controls the wait time from the output of the valid chip select to the output of the valid write enable signal.

Offset Address	Register Name	Total Reset Value
0x010	SMI_BWSTWENR1	0x0000_000F

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

Bits	Access	Name	Description
[31:4]	-	reserved	Reserved.

[3:0]	RW	wstwen	T_{WSTWEN} controls the wait time from the output of the valid chip select to the output of the valid write enable signal. $T_{WSTWEN} = wstwen \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstwen indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BCR1

SMI_BCR1 is the control register of SMI bank 1. It configures the transfer parameters of bank1.

Offset Address	Register Name	Total Reset Value
0x014	SMI_BCR1	0x0030_3020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										reserved	reserved	burstlenwrite		reserved	bmwrite	reserved				burstlenread		reserved	bmread	reserved	reserved	mw		wp	waiten	waitpol	reserved
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	-	reserved	Reserved.
[21]	RW	reserved	Reserved bit set to 1 only.
[20]	RW	reserved	Reserved bit set to 1 only.
[19:18]	RW	burstlenwrite	Burst length that specifies the count of data transferred in a burst write operation in burst mode. 00: Burst 4. 01: Burst 8. 10 or 11: Reserved.
[17]	RW	reserved	Reserved bit set to 0 only.
[16]	RW	bmwrite	Configuration bit of the write mode. 0: Write to the external device in non-burst mode. 1: Write to the external device in burst mode.
[15]	-	reserved	Reserved.
[14]	RW	reserved	Reserved bit set to 0 only.
[13]	RW	reserved	Reserved bit set to 1 only.
[12]	RW	reserved	Reserved bit set to 1 only.

[11:10]	RW	burstlenread	Burst length that specifies the count of data to be transferred in the subsequent burst read in burst mode. 00: Burst 4. 01: Burst 8. 10: Burst 16. 11: Reserved.
[9]	RW	reserved	Reserved bit set to 0 only.
[8]	RW	bmread	Configuration bit of the read mode. 0: Read the external device in non-burst mode. 1: Read the external device in burst mode.
[7]	-	reserved	Reserved.
[6]	RW	reserved	Reserved bit set to 0 only.
[5:4]	RW	mw	Data width of the SMI bank 1 external device. 00: 8 bits. Others: Reserved
[3]	RW	wp	Write-protect bit of the external device. This bit controls whether to write-protect the external device. 0: The external device is not write-protected. For example, the SRAM is writable. 1: The external device is write-protected. for example, the ROM is read-only.
[2]	RW	waiten	SMI bank 1 wait signal enable bit. 0: Disabled. The SMI bank 1 is not controlled by the externally input wait signal. Data is transferred according to the configuration parameters of the internal timing register. 1: Enabled. The SMI bank 1 transfers data according to the externally input wait signal.
[1]	RW	waitpol	Polarity selection bit of the externally input wait signal. 0: Active low. 1: Active high.
[0]	RW	reserved	Reserved bit set to 0 only.

SMI_BSR1

SMI_BSR1 is the status register of SMI bank 1. It displays the current status of bank1.

Offset Address	Register Name	Total Reset Value
0x018	SMI_BSR1	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	waittouterr	External wait timeout error flag. In read operation: 0: No error occurs. 1: An external wait timeout error occurs. In write operation: 0: Invalid. 1: The external wait timeout error flag is cleared.

SMI_BWSTBRDR1

SMI_BWSTBRDR1 is the burst read wait delay control register of SMI bank 1. It controls the subsequent burst read wait delay.

Offset Address	Register Name	Total Reset Value
0x01C	SMI_BWSTBRDR1	0x0000_001F

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

Bits	Access	Name	Description
[31:5]	-	reserved	Reserved.

[4:0]	RW	wstbrd	$T_{WSTBURSTRD}$ controls the burst read wait cycle. For non-burst read, this timing parameter is not used. For burst read, this timing parameter controls the subsequent burst read wait cycle. The initial burst read wait cycle is configured through SMI_BWSTRDR1 $T_{WSTBURSTRD} = wstbrd \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstbrd indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BIDCYR0

SMI_BIDCYR0 is the idle cycle control register of SMI bank 0. It controls the idle cycle between read and write.

Offset Address	Register Name	Total Reset Value
0x0E0	SMI_BIDCYR0	0x0000_000F

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

Bits	Access	Name	Description
[31:4]	-	reserved	Reserved.
[3:0]	RW	idcy	$T_{WSTIDLY}$ controls the idle cycle between read and write. $T_{WSTTIDLY} = idcy \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, idcy indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI.

SMI_BWSTRDR0

SMI_BWSTRDR0 is the read wait cycle control register of SMI bank 0. It controls the read wait cycle.

Offset Address	Register Name	Total Reset Value
0x0E4	SMI_BWSTRDR0	0x0000_001F

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

Bits	Access	Name	Description
[31:5]	-	reserved	Reserved.
[4:0]	RW	wstrd	T_{WSTRD} controls the read wait cycle. For non-burst read, it controls the read wait cycle. For burst read, it controls the initial burst read wait cycle. The subsequent read wait cycles are configured through SMI_BWSTBRDR0. $T_{WSTRD} = wstrd \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstrd indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BWSTWRR0

SMI_BWSTWRR0 is the write wait cycle control register of SMI bank 0. It controls the write wait cycle.

Offset Address	Register Name	Total Reset Value
0x0E8	SMI_BWSTWRR0	0x0000_001F

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

Bits	Access	Name	Description
[31:5]	-	reserved	Reserved.
[4:0]	RW	wstwr	T_{WSTWR} controls the write wait cycle. $T_{WSTWR} = wstwr \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstwr indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BWSTOENR0

SMI_BWSTOENR0 is the read enable assertion delay control register of SMI bank 0. It controls the wait time from the output of the valid chip select signal to the output of the valid read enable signal.

Offset Address	Register Name	Total Reset Value
0x0EC	SMI_BWSTOENR0	0x0000_000F

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

Bits	Access	Name	Description
[31:4]	-	reserved	Reserved.
[3:0]	RW	wstoen	T_{WSTOEN} controls the wait time from the output of the valid chip select signal to the output of the valid read enable signal. $T_{WSTOEN} = wstoen \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstoen indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BWSTWENR0

SMI_BWSTWENR0 is the write enable assertion delay control register of SMI bank 0. It controls the wait time from the output of the valid chip select to the output of the valid write enable signal.

Offset Address	Register Name	Total Reset Value
0x0F0	SMI_BWSTWENR0	0x0000_000F

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

Bits	Access	Name	Description
[31:4]	-	reserved	Reserved.

[3:0]	RW	wstwen	T_{WSTWEN} controls the wait time from the output of the valid chip select to the output of the valid write enable signal. $T_{WSTWEN} = wstwen \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstwen indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_BCR0

SMI_BCR0 is the control register of SMI bank 0. It configures the transfer parameters of bank0.

Offset Address	Register Name	Total Reset Value
0x0F4	SMI_BCR0	0x0030_3000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved										reserved	reserved	burstlenwrite		reserved	bmwrite	reserved				burstlenread		reserved	bmread	reserved	reserved	mw		wp	waiten	waitpol	reserved
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	-	reserved	Reserved.
[21]	RW	reserved	Reserved bit set to 1 only.
[20]	RW	reserved	Reserved bit set to 1 only.
[19:18]	RW	burstlenwrite	Burst length that specifies the count of data to be transferred in a burst write operation in burst mode (see Figure 4-5). 00: Burst 4. 01: Burst 8. 10 or 11: Reserved.
[17]	RW	reserved	Reserved bit set to 0 only.
[16]	RW	bmwrite	Configuration bit of the write mode. 0: Write to the external device in non-burst mode. 1: Write to the external device in burst mode.
[15]	-	reserved	Reserved.
[14]	RW	reserved	Reserved bit set to 0 only.
[13]	RW	reserved	Reserved bit set to 1 only.
[12]	RW	reserved	Reserved bit set to 1 only.

[11:10]	RW	burstlenread	Burst length that specifies the count of data to be transferred in the subsequent burst read operation in burst mode. 00: Burst 4. 01: Burst 8. 10: Burst 16. 11: Reserved.
[9]	RW	reserved	Reserved bit set to 0 only.
[8]	RW	bmread	Configuration bit of the read mode. 0: Read the external device in non-burst mode. 1: Read the external device in burst mode.
[7]	-	reserved	Reserved.
[6]	RW	reserved	Reserved bit set to 0 only.
[5:4]	RW	mw	Data width of the SMI bank 0 external device. 00: 8 bits. Others: Reserved.
[3]	RW	wp	Write-protect bit of the external device. This bit controls whether to write-protect the external device. 0: The external device is not write-protected. For example, the external SRAM is writable. 1: The external device is write-protected. For example, the external ROM is read-only.
[2]	RW	waiten	SMI bank 0 wait signal enable bit. 0: Disabled. The SMI bank 0 is not controlled by the externally input wait signal. Data is transferred according to the configuration parameters of the internal timing register. 1: Enabled. The SMI bank 0 transfers data according to the externally input wait signal.
[1]	RW	waitpol	Polarity selection bit of the externally input wait signal. 0: Active low. 1: Active high.
[0]	RW	reserved	Reserved bit set to 0 only.

SMI_BSR0

SMI_BSR0 is the status register of SMI bank 0. It displays the current status of bank0.

Offset Address	Register Name	Total Reset Value
0x0F8	SMI_BSR0	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	waittouterr	External wait timeout error flag. In read operation: 0: No error occurs. 1: An external wait timeout error occurs. In write operation: 0: Invalid. 1: The external wait timeout error flag is cleared.

SMI_BWSTBRDR0

SMI_BWSTBRDR0 is the burst read wait delay control register of SMI bank 0. It controls the subsequent burst read wait delay.

Offset Address	Register Name	Total Reset Value
0x0FC	SMI_BWSTBRDR0	0x0000_001F

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

Bits	Access	Name	Description
[31:5]	-	reserved	Reserved.

[4:0]	RW	wstbrd	$T_{WSTBURSTRD}$ controls the burst read wait cycle. For non-burst read, this timing parameter is not used. For burst read, this timing parameter controls the subsequent burst read wait cycles. The initial burst read wait cycle is configured through SMI_BWSTRDR0. $T_{WSTBURSTRD} = wstbrd \times \left(\frac{1}{f_{SMICLK}} \right)$ Where, wstbrd indicates the number of working clock cycles of the system configuration SMI controller; f_{SMICLK} indicates the frequency of the working clock of the SMI controller.

SMI_SR

SMI_SR is the SMI asynchronous wait status register that displays the asynchronous wait status.

Offset Address	Register Name	Total Reset Value
0x200	SMI_SR	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RO	waitstatus	SMI asynchronous wait status bit. 0: The wait signal is invalid. 1: The wait signal is valid.

Note: This status register displays the asynchronous wait status in asynchronous wait mode only.

SMI_CR

SMI_CR is the SMI working clock configuration register that configures the relationship between the working clock of the SMI controller and the bus clock of the system.

Offset Address	Register Name	Total Reset Value
0x204	SMI_CR	0x0000_0001

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

Bits	Access	Name	Description
[31:3]	-	reserved	Reserved.
[2:1]	RW	memclkratio	Configuration bit of the working clock of the SMI controller. 00: The frequency of the working clock of the SMI controller is equal to that of the system bus clock. 01: The frequency of the working clock of the SMI controller is equal to the frequency of the system bus clock divided by 2. 10 or 11: Reserved.
[0]	RW	reserved	Reserved bit set to 1 only.

4.3 NAND Flash Controller

4.3.1 Overview

The NAND flash controller (NANDC) connects to an external NAND flash to access data through its memory controller interfaces.

4.3.2 Features

The NANDC has the following features:

- Provide 2 KB (2048 bytes + 128 bytes) on-chip buffer to improve the read speed
- Supports two chip select (CS) signals and one ready/busy signal
- Supports 8-bit data-bus NAND flash interfaces
- Supports the NAND boot function and the NAND flash with the page size of 2 KB or 4 KB. In addition, the NANDC can boot from the NAND flash corresponding to CS 0.
- Supports 512-byte hamming error correcting code (ECC) check and 1-bit error correction. For the MLC component, the NANDC supports 4-/8-byte (512 bytes) error check and correction. It can also enable or disable ECC check and 1-bit error correction.
- Reports interrupts when errors occur during reading, writing, erasing, programming, and ECC checking
- Reads and writes data with variable length

- Flexibly configures the commands issued by the controller to support various NAND flash command operations (including cache read and write)
- Supports break in the process of reading and writing the NAND flash, thus releasing the shared bus
- Operates two NAND flash memories alternatively to improve the efficiency
- Supports write protection for the NAND flash. The write-protected addresses are configurable.
- Supports lock and lock_down modes and enabling or disabling the functions of flash lock and flash global lock. By default, the functions are enabled. The NANDC always reports operation error interrupts if the locked addresses are written.
- Reads data from the NAND flash in enhanced data out (EDO) mode

4.3.3 Description of NANDC Interfaces

4.3.3.1 Description of Interface Signals

Table 4-13 shows the signals of NANDC interfaces.

Table 4-13 Signals of NANDC interfaces

Signal	Direction	Description	Corresponding Pin
NF_RB	I	NAND flash status indicator signal. 1: idle 0: busy	NFRB
NF_CSN0	O	First CS signal of the NAND flash, active low.	NFCS0N
NF_CSN1	O	Second CS signal of the NAND flash, active low.	NFCS1N
NF_REN	O	Read enable signal of the NAND flash, active low.	NFOEN
NF_WEN	O	Write enable signal of the NAND flash, active low.	EBIWEN
NF_CLE	O	Command latch signal of the NAND flash.	NFCLE
NF_ALE	O	Address latch signal of the NAND flash.	NFALE
NF_ADNUM[1:0]	I	Number of addresses sent to the NAND flash by the NANDC during booting. 00: three address cycles 01: four address cycles 10: five address cycles 11: six address cycles	EBIADR18/EBIADR17

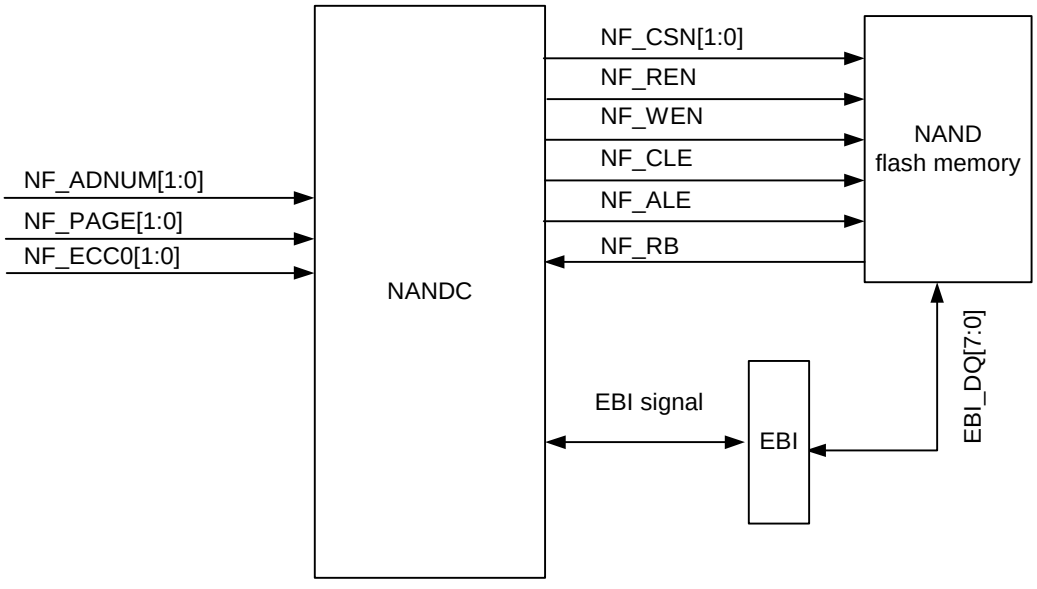
Signal	Direction	Description	Corresponding Pin
NF_PAGE[1:0]	I	Page size of the NAND flash during booting. 01: 2 KB 10: 4 KB Others: reserved	EBIADR20/EBIADR19
NF_ECC0[1:0]	I	ECC mode during booting. 00: non-ECC mode 01: 1-bit ECC mode 10: 4-bit ECC mode 11: 8-bit ECC mode By default, the 1-bit correction mode is enabled.	EBIADR16/EBIADR15
EBI_DQ[7:0]	IO	Data address bus of the NAND flash interface. When an 8-bit NAND flash is accessed, only the lower eight bits are valid.	EBI_DQ0–EBI_DQ7

4.3.4 Function Description

4.3.4.1 Block Diagram of NANDC Interfaces

The Hi3515 provides two CS signals and one ready/busy signal, so it can connect to the NAND flash conveniently. If there are two ready/busy signals, they are connected to NFRB in wire-ANDed mode.

Figure 4-7 Block diagram of NANDC interfaces



4.3.4.2 Function Principle

The data of the NAND flash is stored by blocks or by pages. Each block contains several pages. Before writing data to the NAND flash, you must erase the memory by blocks. Then, you should read and write to the memory by pages.

The commands for operating the NAND flash vary according to manufacturers. That is, the specific commands are subject to component manuals.

A typical read operation is performed as follows:

Step 1 Send the 0x00 command to the NAND flash.

Step 2 Send the read start address.

The start address consists of the internal page address, page address, and block address. For details, see the NAND flash manuals provided by related manufacturers.

Step 3 Send the read acknowledgement command 0x30.

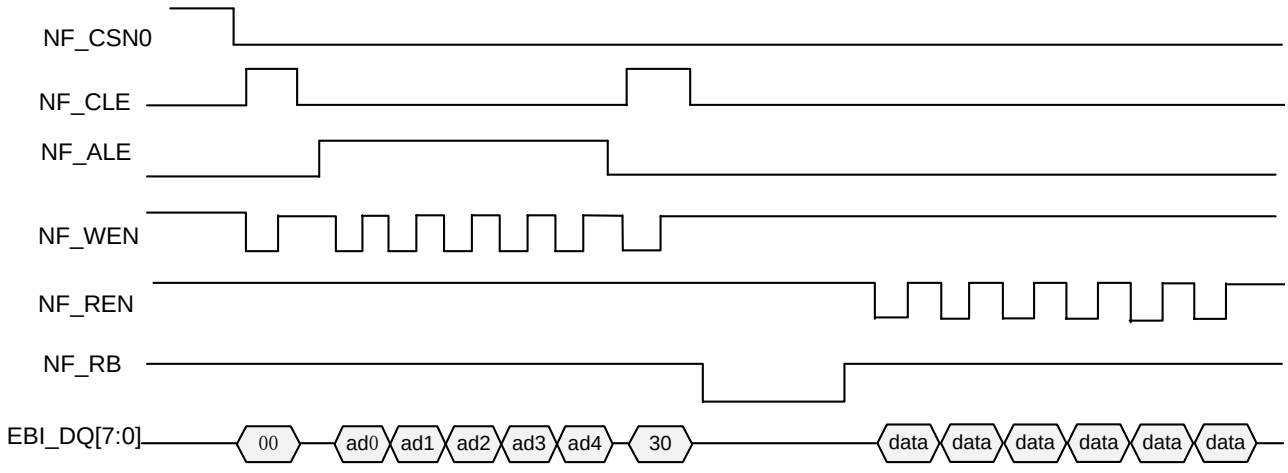
In this case, the NAND flash pulls the RB signal low. This indicates that the NAND flash is being read. About 25 μs later, the RB signal becomes high. This indicates that data of the NAND flash has been prepared.

Step 4 Read data from the NAND flash by enabling the NF_REN signal.

----End

Figure 4-8 shows the typical timing when the NANDC reads the data of a page size from the NAND flash.

Figure 4-8 Typical timing when the NANDC reads the data of a page size from the NAND flash



A typical programming operation (writing data) is performed as follows:

Step 1 Send the programming command 0x80 to the NAND flash.

Step 2 Send the start address where data is written.

The start address consists of the internal page address, page address, and block address. For details, see the NAND flash manuals provided by related manufacturers.

Step 3 Write data to the internal buffer of the NAND flash.

Step 4 Send the programming acknowledgement command 0x10.

In this case, the NAND flash pulls the RB signal low. This indicates that the NAND flash is being programmed. About 200 ms later, the RB signal becomes high. This indicates that the programming operation is complete.

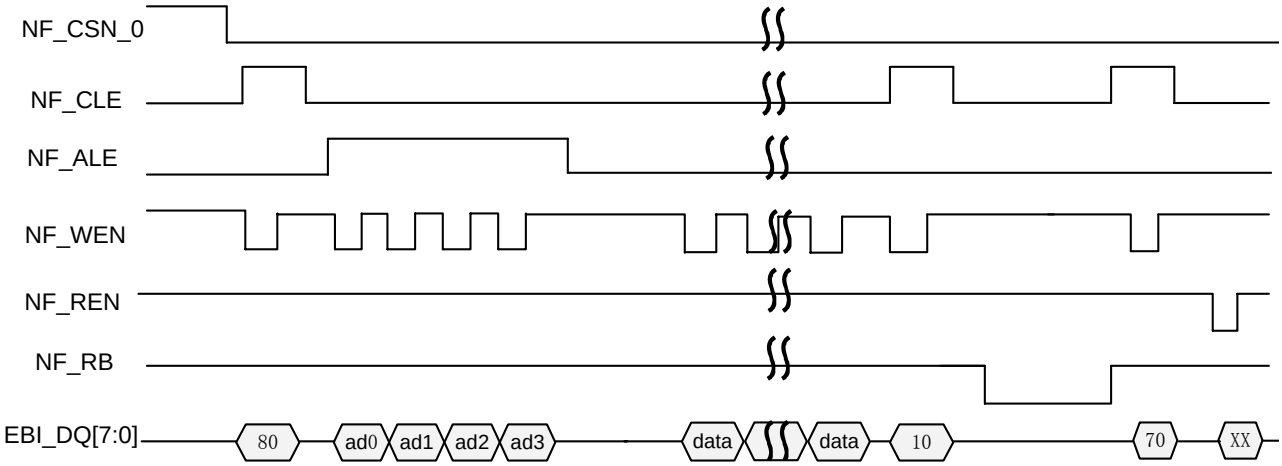
Step 5 Read the status data by sending the read status command 0x70.

The status data indicates whether the programming operation is successful.

----End

Figure 4-9 shows the timing when the NANDC starts to be programmed.

Figure 4-9 Timing when the NANDC starts to be programmed



4.3.5 Operating Mode

Pin Multiplexing Configuration

NFRB is multiplexed with GPIO0_7. If a peripheral has two ready/busy signals, they are connected to NF_RB in wire-ANDed mode.

NFCS1N is multiplexed with GPIO0_6. If a peripheral has two CS signals, the IO config register reg43 must be configured. That is, NFCS1N should be selected.

Clock Gating

When the NAND flash is not used, you can disable the working clock of the NANDC. To disable the clock, do as follows:

Step 1 Read NFC_STATUS[nfc_ready] of the NANDC.

Step 2 If NFC_STATUS[nfc_ready] is 1 and the NAND flash is not read and written by the software, go to Step 3; otherwise, go to Step 1.

Step 3 Write 1 to the system register SC_PERDIS[nandcclkdis] to disable the clock.

----End

Soft Reset

After the NANDC is enabled by writing data to the NFC_OP, if NFC_STATUS[nfc_ready] is changed to 0 and cannot be changed to 1 for a long period, it indicates that the NANDC is abnormal and a soft reset operation is required. The maximum duration depends on the NAND flash. For the single-level cell (SLC) flash memory, it is less than 4 ms; for the multi-level cell (MLC) memory, it is less than 11 ms.

To execute soft reset on the NANDC, write 1 to SC_PERCTRL10 [nadc_hrst]. After reset, each configuration register is reset to its default value. Therefore, these registers must be initialized again after reset. In addition, the reset command must be sent to the NAND flash after soft reset (if it supports the reset operation), thus ensuring its reliability.

NOTE

The NANDC provides the lock function. After this function is enabled, it can be disabled only after hardware is reset.

Boot Configuration Pins

The NANDC supports the NAND boot function and the memory with the page size of 2 KB or 4 KB. The NANDC can boot from the NAND flash corresponding to CS 0 only.

After reset, the boot mode of the NANDC depends on the levels of the NANDC boot configuration pins. When the NANDC is reset, the levels of the boot configuration pins are sampled. After sampling, the operating status of the NANCD is not affected by the levels. Table 4-14 shows the boot configuration pins.

Table 4-14 Boot configuration pins

Pin	Direction	Description
NF_NUM[1:0]	I	Number of addresses sent to the NAND flash by the NANDC during booting. 00: three address cycles 01: four address cycles 10: five address cycles 11: six address cycles
NF_PAGE[1:0]	I	Page size of the NAND flash during booting. 01: 2 KB 10: 4 KB Others: reserved

Pin	Direction	Description
NF_ECC[1:0]	I	ECC mode during booting. 00: non-ECC mode 01: 1-bit ECC mode 10: 4-bit ECC mode 11: 8-bit ECC mode

Boot Mode

By default, the NANDC is in boot mode and can boot from the NAND flash corresponding to CS 0 only. The features of the boot mode are as follows:

- The data ranging from 0x00_0000 to 0x01_FFFF can be read through the CPU. The total size of the address space is 128 KB.
- When the NANDC boots from the NAND flash, the NANDC automatically sends a command to read the corresponding page of the NAND flash based on the addresses read by the CPU and then returns the corresponding data.
- Data cannot be written to the internal buffer through the CPU.
- The boot configuration pins must be configured properly according to the models of the connected NAND flash.

Normal Mode

The NANDC is switched to the normal mode when NFC_CON[op_mode] is set to 1. In this mode, the operations such as erasing, programming, and reading can be performed on the NAND flash through the CPU.

Setting NAND Flash Addresses

The NANDC does not translate addresses. According to the number of addresses set by the configuration register, the NANDC sends the values of low-bit and high-bit address registers to NAND flash. Hence, the software should translate the address of the CPU into the address of the NAND flash and then write it to the address register. The requirements for setting the address of each flash memory are subject to the user manuals provided by the manufacturers of NAND flash memories.

Table 4-15 lists the requirements for setting the addresses of the Samsung K9F2G08U0M memory. Its capacity is 256 MB x 8 bits and its page size is 2 KB. A0 to A11 indicate internal page addresses (column addresses) and A12 to A27 indicate page addresses (row addresses).

Table 4-15 K9F2G08U0M addresses

Cycle	IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7
1st cycle	A0	A1	A2	A3	A4	A5	A6	A7
2nd cycle	A8	A9	A10	A11	0	0	0	0
3rd cycle	A12	A13	A14	A15	A16	A17	A18	A19

Cycle	IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7
4th cycle	A20	A21	A22	A23	A24	A25	A26	A27

Table 4-16 lists the requirements for setting the addresses of the Samsung K9GAG08X0M memory. Its capacity is 2 GB x 8 bits and its page size is 4KB. A0 to A12 indicate internal page addresses (column addresses) and A13 to A31 indicate page addresses (row addresses).

Table 4-16 K9GAG08X0M addresses

Cycle	IO0	IO1	IO2	IO3	IO4	IO5	IO6	IO7
1st cycle	A0	A1	A2	A3	A4	A5	A6	A7
2nd cycle	A8	A9	A10	A11	A12	0	0	0
3rd cycle	A13	A14	A15	A16	A17	A18	A19	A20
4th cycle	A21	A22	A23	A24	A25	A26	A27	A28
5th cycle	A29	A30	A31	0	0	0	0	0

Address Mapping

In normal mode, the address mapping of the NANDC is as follows:

- The base address of the internal buffer of the NANDC is 0x7000_0000.
- The base address of the internal register area of the NANDC is 0x1000_0000.

Operation Commands

The NAND flash memories provide certain advanced commands. Table 4-17 shows the common commands for operating NAND flash memories.

Table 4-17 Common commands for operating the NAND flash memories

Function	1st Cycle	2nd Cycle	Remarks
READ	00H	30H	–
PROGRAM	80H	10H	–
BLOCK ERASE	60H	D0H	–
READ ID	90H	–	–
READ STATUS	70H	–	–
RESET	FFH	–	–

Data Storage Structure

The size of the internal buffer of the NANDC is (2048 + 128) bytes. This section describes the data storage structures in the NANDC buffer. The data is read from or written to the NAND flash.

- 1-bit ECC Mode
  - Page Size of (512 + 16) Bytes

    The 512-byte valid data is stored in the addresses of the buffer ranging from 0x000 to 0x1FF; the data of the spare area is stored in the addresses ranging from 0x7FF to 0x83F, as shown in Figure 4-10.
  - Page Size of 2 KB (2048 + 64 Bytes)

    The 2048-byte valid data is stored in the addresses of the buffer ranging from 0x000 to 0x7FF; the 64-byte spare data is stored in the addresses ranging from 0x800 to 0x80F.

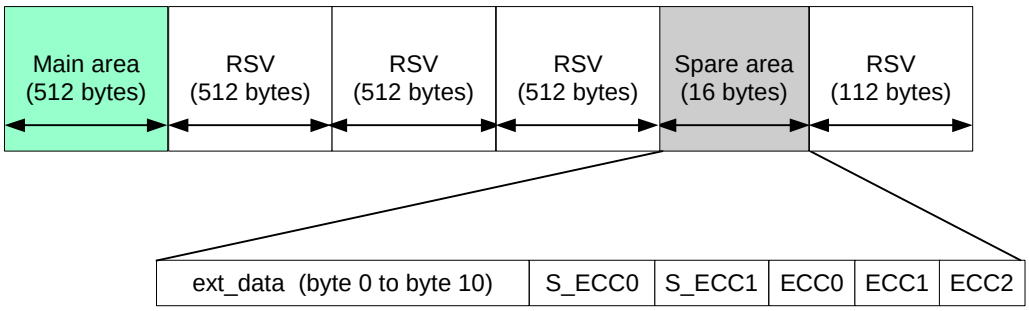
    The main areas of byte 0 to byte 511 map to the spare areas of the addresses ranging from 0x800 to 0x80F. The rest may be deduced by analogy.

    The data structures of spare areas are the same as those with the page size of (512 + 16) bytes, as shown in Figure 4-11.

    When ECC mode is enabled, the data written to the NAND flash is automatically stored in the format that each main area is located next to its corresponding spare area, as shown in Figure 4-12.
  - Page Size of 4 KB (4096 + 128 Bytes)

    For the memory with the page size of 4 KB, the operation must be performed twice. The structures for reading and writing data each time are the same as those in Figure 4-11.

Figure 4-10 Data storage structure of the NAND flash with the page size of (512 + 16) bytes in 1-bit ECC mode



NOTE

- ECC0, ECC1, and ECC2: ECC code for main area data
- S_ECC0 and S_ECC1: ECC code for RSV (byte 0 to byte 10) data
- ext_data: extended data area

Figure 4-11 Data storage structure of the NAND flash with the page size of 2 KB (2048 + 64) bytes

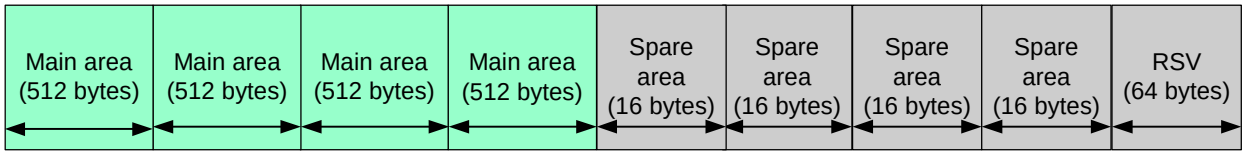
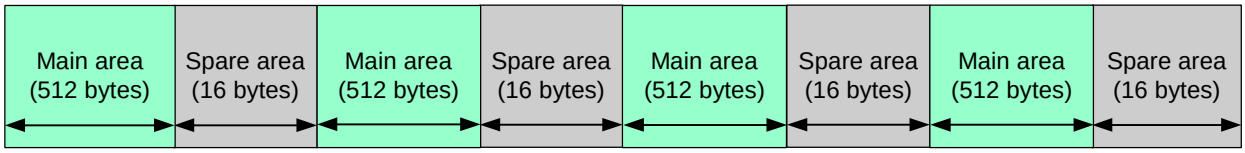


Figure 4-12 Data structure after data is automatically stored in main areas and their corresponding spare areas



NOTE

In boot mode, if the 1-bit ECC function is enabled, the data must be stored in the NAND flash in the structure as shown in Figure 4-11.

- 4-bit ECC Mode
  - Page Size of 2 KB (2048 + 64 Bytes)

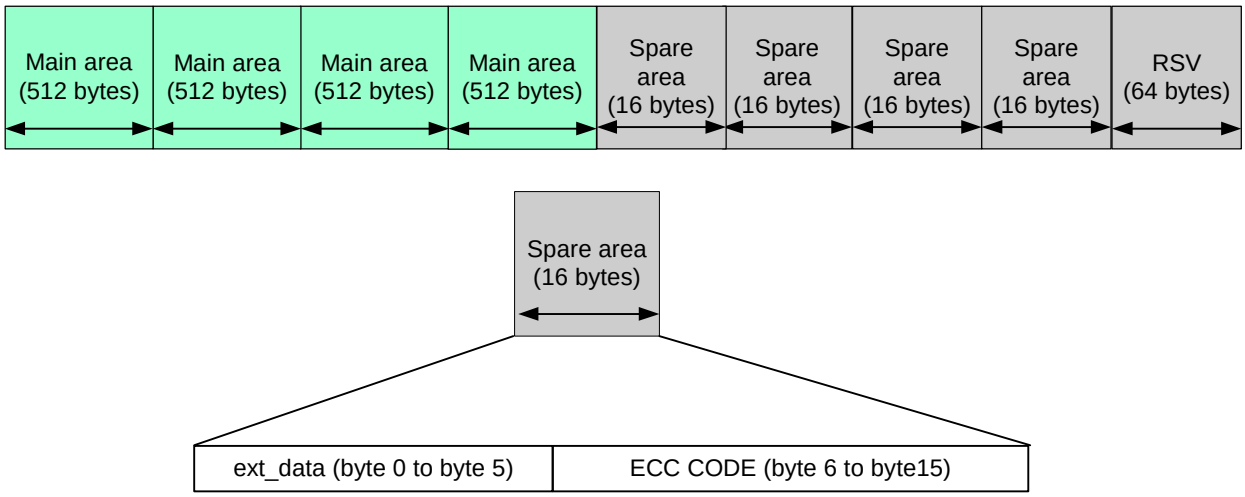
    The 2048-byte valid data is stored in the addresses of the buffer ranging from 0x000 to 0x7FF; the 64-byte spare data is stored in the addresses ranging from 0x800 to 0x83F, as shown in Figure 4-13.

    The main areas of byte 0 to byte 511 map to the spare areas of byte 2048 to byte 2063. The rest may be deduced by analogy.

    When ECC check is enabled, the data written to the NAND flash is automatically stored in the format that each main area is located next to its corresponding spare area, as shown in Figure 4-12.
  - Page Size of 4KB (4096 + 128 Bytes)

    For the memory with the page size of 4 KB, the operation must be performed twice. The structures for reading and writing data each time are the same as those in Figure 4-13.

Figure 4-13 Data storage structure of the NAND flash with the page size of 2 KB (2048 + 64) bytes in 4-bit ECC mode



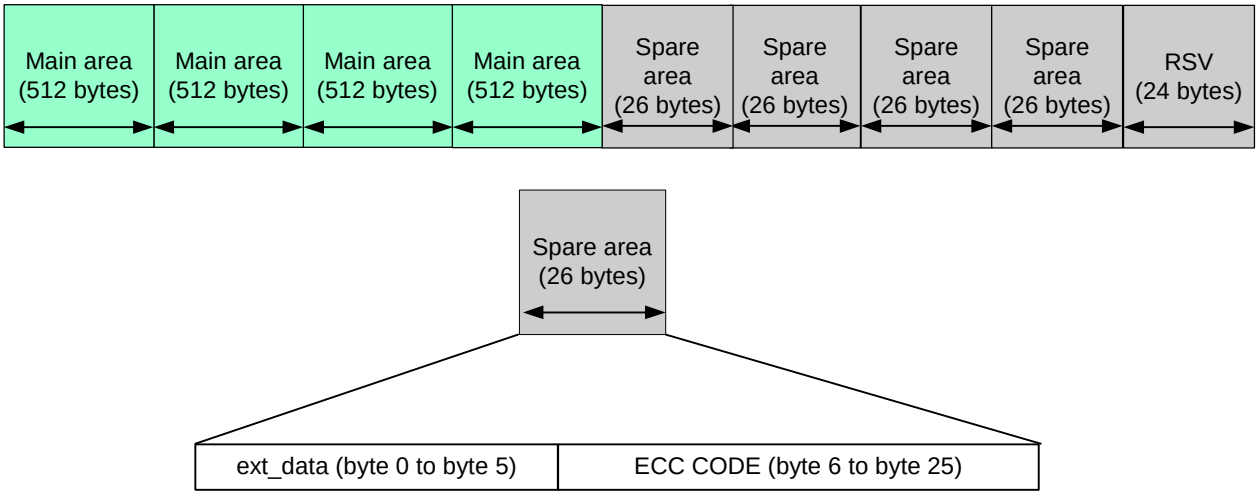
- 8-Bit ECC Mode

  The following shows the storage structure of the data in the internal buffer of the NANDC. The buffer size is (2048 + 26 x 4) bytes. For the memory with the page size of 4 KB, the operation must be performed twice, as shown in Figure 4-14.

  When ECC check is enabled, the data written to the NAND flash is automatically stored in the format that each main area is located next to its corresponding spare area, as shown in Figure 4-12.

  For the Toshiba memory, its storage space is (4096 + 218) bytes; therefore, 10 bytes (byte 209 to byte 218) are left.

Figure 4-14 Data storage structure of the NAND flash with the page size of (2048 + 26 x 4) bytes in 8-bit ECC mode



Initialization

The initialization operation is performed as follows:

Step 1 Write 1 to NFC_CON[op_mode] to enter the normal mode, set NFC_CON[page_size] based on the page size of the connected memory, and then set check and correction modes based on the number of CS signals and ready/busy signals of the connected memory and the configuration of NFC_CON[ecc_type].

Step 2 Write to NFC_PWIDTH based on the timing requirements for the connected memory.

Step 3 In query mode, write to the interrupt enable register NFC_INTEN to mask all the interrupts; in interrupt mode, enable the interrupt op_done only and mask other interrupts.

----End

Erasing the NAND Flash

The NAND flash is erased as follows:

Step 1 Write the programming page address to NFC_ADDRL and NFC_ADDRH and write the erase command 0x0070_D060 to NFC_CMD.

Step 2 Write 0x369 to NFC_OP and enable the NANDC to erase the NAND flash. Assume that the NAND flash chip needs three addresses and the operation is performed on CS 0.

Step 3 In query mode, check the value of NFC_STATUS[nfc_ready]. If it is 1, go to Step 4; otherwise, continue to query. In interrupt mode, check the value of NFC_INTS[op_done]. If it is 1, go to Step 4.

Step 4 Read NFC_STATUS[nf_status] to check whether the erase operation is successful.

----End

Programming the NAND Flash with the Page Size of 4 KB

The NAND flash is programmed as follows:

Step 1 Write the first half of 4-KB data (including 2048-byte data in main areas and 104-byte data in spare areas) to the buffer of the NANDC through the CPU. If ecc_type is 2'b00, write to NFC_DATA_NUM to set the number of data segments to be written.

Step 2 Write the page address of the NAND flash to NFC_ADDRL and NFC_ADDRH and write the programming command 0x0070_1080 to NFC_CMD through the CPU.

Step 3 Write 0x570 to NFC_OP and enable the NANDC to write to the NAND flash through the CPU. Assume that the NAND flash chip needs five addresses and the operation is performed on CS 0.

Step 4 In query mode, check the value of NFC_STATUS[nfc_ready] until the flag is 1, and then go to Step 5; in interrupt mode, check the value of NFC_INTS[op_done]. If the op_done interrupt is generated, go to Step 5.

Step 5 Write the last half of 4-KB data (including 2048-byte data in main areas and 104-byte data in spare areas) to the buffer of the NANDC through the CPU. Write 0x0d to NFC_OP and enable the NANDC to send a programming command to the NAND flash.

Step 6 In query mode, check the value of NFC_STATUS[nfc_ready] until the flag is 1, and then go to Step 7; in interrupt mode, check the value of NFC_INTS[op_done]. If the op_done interrupt is generated, go to Step 7.

Step 7 Read NFC_STATUS [nf_status] to check whether the programming operation is successful.

----End

Reading Page-Sized Data from the NAND Flash with the Page Size of 4 KB

Data is read as follows:

Step 1 Write the page address of the NAND flash to NFC_ADDRL and NFC_ADDRH and write the read command 0x3000 to NFC_CMD through the CPU. If ecc_type is 2'b00, write to NFC_CMD to set the number of data segments to be read.

Step 2 Write 0x56E to NFC_OP to enable the NANDC to read data. Assume that the NAND flash chip needs five addresses and the operation is performed on CS 0.

Step 3 In query mode, check the value of NFC_STATUS[nfc_ready] until the flag is 1, and then go to Step 4; in interrupt mode, check the value of NFC_INTS[op_done]. If the op_done interrupt is generated, go to Step 4.

Step 4 Read NFC_STATUS through the CPU to check whether uncorrectable errors occur.

Step 5 Read data from the buffer of the NANDC through the CPU, and then write the data to the memory.

Step 6 Write 0x02 to NFC_OP to enable the NANDC to continue to read the last half of data with the page size of 4 KB.

Step 7 In query mode, check the value of NFC_STATUS[nfc_ready] until the flag is 1, and then go to Step 8; in interrupt mode, check the value of NFC_INTS[op_done]. If the op_done interrupt is generated, go to Step 8.

Step 8 Read NFC_STATUS through the CPU to check whether uncorrectable errors occur.

Step 9 Read data from the buffer of the NANDC through the CPU, and then write the data to the memory.

----End

Other Precautions

Other precautions that should be taken are as follows:

- Operation commands supported by the NAND flash memories vary according to manufacturers. Therefore, the command register NFC_CMD must be configured properly according to memory manuals. Also, the NAND flash memories with different capacities require different number of address cycles. Thus the address_cycles field of NFC_OP must be configured according to memory manuals. In addition, the timings supported by memories are different. Hence, the read-write pulse width register NFC_PWIDTH and operation interval register NFC_OPIDLE must be configured according to memory manuals.
- After configuring related registers and buffer, write to NFC_OP to enable the NANDC to read and write to the flash memory. After that, do not write related registers; otherwise, the NANDC or flash memory may work improperly.

- After enabling the NAND flash to be read and written by writing NFC_STATUS, do not read and write to the buffer of the NANDC when the flag NFC_STATUS[nfc_ready] is 0. Otherwise, error data may be returned.

4.3.6 Register Summary

Table 4-18 lists the NANDC registers.

Table 4-18 Summary of NANDC registers (base address: 0x1000_0000)

Offset Address	Register	Description	Page
0x00	NFC_CON	NANDC configuration register	4-74
0x04	NFC_PWIDTH	Read/Write pulse width configuration register	4-75
0x08	NFC_OPIDLE	Operation interval configuration register	4-76
0x0C	NFC_CMD	Command word configuration register	4-76
0x10	NFC_ADDRL	Low-bit address configuration register	4-77
0x14	NFC_ADDRH	High-bit address configuration register	4-77
0x18	NFC_DATA_NUM	Configuration register for the number of data segments to be read and written	4-77
0x1C	NFC_OP	Operation register	4-78
0x20	NFC_STATUS	Status register	4-80
0x24	NFC_INTEN	Interrupt enable register	4-81
0x28	NFC_INTS	Interrupt status register	4-82
0x2C	NFC_INTCLR	Interrupt clear register	4-83
0x30	NFC_LOCK	Lock address configuration register	4-85
0x34	NFC_LOCK_SA0	Configuration address of lock start address 0	4-85
0x38	NFC_LOCK_SA1	Configuration address of lock start address 1	4-86
0x3C	NFC_LOCK_SA2	Configuration address of lock start address 2	4-87
0x40	NFC_LOCK_SA3	Configuration address of lock start address 3	4-87
0x44	NFC_LOCK_EA0	Configuration address of lock end address 0	4-88
0x48	NFC_LOCK_EA1	Configuration address of lock end address 1	4-88
0x4C	NFC_LOCK_EA2	Configuration address of lock end address 2	4-89
0x50	NFC_LOCK_EA3	Configuration address of lock end address 3	4-90
0x54	NFC_EXPCMD	Extended page command register	4-90
0x58	NFC_EXBCMD	Extended block command register	4-91

Offset Address	Register	Description	Page
0x5C	NFC_ECC_TEST	ECC test register	4-91

4.3.7 Register Description

NFC_CON

NFC_CON is the NANDC configuration register.

Offset Address	Register Name	Total Reset Value
0x00	NFC_CON	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																			edo_en	ecc_type		ext_data_ecc_en	protection_en	rb_sel		cs_ctrl	ecc_en	reserved		pagesize	op_mode
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	1	0	1	?	?	0	0	0

Bits	Access	Name	Description
[31:13]	-	reserved	Reserved.
[12]	RW	edo_en	NAND flash data read/write enable in EDO mode. 0: normal mode 1: EDO mode This function must be used according to the requirements for specific memory.
[11:10]	RW	ecc_type	ECC mode selection. 00: non-ECC mode 01: 1-bit ECC mode 10: 4-bit ECC mode 11: 8-bit ECC mode The reset value depends on the pins EBIADR16 and EBIADR15.
[9]	RW	ext_data_ecc_en	Check or correction enable for the extended data area. 0: disabled 1: enabled Note: This bit is valid only when ecc_en is 1.
[8]	RW	protection_en	Register read- and write-protection enable. 0: disabled 1: enabled

[7]	-	reserved	Reserved.
[6]	RW	cs_ctrl	CS control. 0: When the NAND flash is busy, keep CS signal 0 unchanged. 1: When the NAND flash is busy, set the CS signal to 1. This mode maps to the cs do not care mode of the NAND flash.
[5]	RW	ecc_en	ECC enable. 0: disabled 1: enabled
[4:3]	-	reserved	Reserved.
[2:1]	RW	pagesize	Page size of the NAND flash. 01: 2 KB 10: 4 KB Others: reserved The reset value depends on the pin NFC_PAGE_SIZE.
[0]	RW	op_mode	Operating mode of the NANDC. 0: boot mode 1: normal mode

NFC_PWIDTH

NFC_PWIDTH is the read/write pulse width configuration register.

Offset Address	Register Name	Total Reset Value
0x04	NFC_PWIDTH	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																				rw_hcnt				r_lcnt				w_lcnt			
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	1	0	1	0	1	0	1	0	1	0	1

Bits	Access	Name	Description
[31:12]	-	reserved	Reserved.
[11:8]	RW	rw_hcnt	High-level width of the read/write signal of the NAND flash. 0x0–0xF: 1–16 clock cycles
[7:4]	RW	r_lcnt	Low-level width of the read signal of the NAND flash. 0x0–0xF: 1–16 clock cycles
[3:0]	RW	w_lcnt	Low-level width of the write signal of the NAND flash. 0x0–0xF: 1–16 clock cycles

NFC_OPIDLE

NFC_OPIDLE is the operation interval configuration register.

Offset Address	Register Name	Total Reset Value
0x08	NFC_OPIDLE	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								frb_wait				cmd1_wait				addr_wait				write_data_wait				cmd2_wait				frb_idle			
Reset	?	?	?	?	?	?	?	?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Access	Name	Description
[31:24]	-	reserved	Reserved.
[23:20]	RW	frb_wait	A period of delay later after a read/write command is sent, the read signal is detected to check whether it becomes high. The number of delay cycles is frb_wait x 8.
[19:16]	RW	cmd1_wait	Number of wait cycles after command 1 is sent. 0x0–0xF: 1–16 clock cycles
[15:12]	RW	addr_wait	Number of wait cycles after the address is sent. 0x0–0xF: 1–16 clock cycles
[11:8]	RW	write_data_wait	Number of wait cycles after data is written. 0x0–0xF: 1–16 clock cycles
[7:4]	RW	cmd2_wait	Number of wait cycles after command 2 is sent. 0x0–0xF: 1–16 clock cycles
[3:0]	RW	frb_idle	A read signal can be sent a period of delay later after the read signal of the NAND flash becomes high. The number of delay cycles is frb_idle x 8.

NFC_CMD

NFC_CMD is the command word configuration register.

Offset Address	Register Name	Total Reset Value
0x0C	NFC_CMD	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								read_status_cmd								cmd2								cmd1							

Reset	?	?	?	?	?	?	?	?	0	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	-	reserved	Reserved.
[23:16]	RW	read_status_cmd	Read status command word.
[15:8]	RW	cmd2	Command 2 that is sent to the NAND flash by the NANDC.
[7:0]	RW	cmd1	Command 1 that is sent to the NAND flash by the NANDC.

NFC_ADDRL

NFC_ADDRL is the low-bit address configuration register.

Offset Address	Register Name	Total Reset Value
0x10	NFC_ADDRL	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	addr_l	Low 32-bit address of the NAND flash.

NFC_ADDRH

NFC_ADDRH is the high-bit address configuration register.

Offset Address	Register Name	Total Reset Value
0x14	NFC_ADDRH	-

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

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RW	addr_h	High 16-bit address of the NAND flash.

NFC_DATA_NUM

NFC_DATA_NUM is the configuration register for the number of data segments to be read and written.

Offset Address	Register Name	Total Reset Value
0x18	NFC_DATA_NUM	-

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

Bits	Access	Name	Description
[31:12]	-	reserved	Reserved.
[11:0]	RW	nfc_data_num	Number of data segments to be read and written randomly by the NANDC. The maximum value is 2152 (2048 + 26 x 4) bytes. **Note**: These bits are valid only when ecc_type is 1.

NFC_OP

NFC_OP is the operation register.

Offset Address	Register Name	Total Reset Value
0x1C	NFC_OP	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																				address_cycles			nf_cs		cmd1_en	addr_en	write_data_en	cmd2_en	wait_ready_en	read_data_en	read_status_en
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:12]	-	reserved	Reserved.
[11:9]	RW	address_cycles	Number of address cycles sent to the NAND flash. The reset value depends on the pins EBIADR17 and EBIADR18. That is, the reset value is the values of EBIADR17 and EBIADR17 plus 3.
[8:7]	RW	nf_cs	NAND flash selection. 00: CS 0 01: CS 1 Others: reserved
[6]	RW	cmd1_en	Command 1 send enable. 0: disabled 1: enabled

[5]	RW	addr_en	Write address to the NAND flash enable. 0: disabled 1: enabled
[4]	RW	write_data_en	Write data to the NAND flash enable. 0: disabled 1: enabled **Note**: read_data_en and write_data_en cannot be 1 at the same time.
[3]	RW	cmd2_en	Command 2 send enable. 0: disabled 1: enabled
[2]	RW	wait_ready_en	Wait ready/busy signal to be high enable. 0: disabled 1: enabled
[1]	RW	read_data_en	Enable bit of starting the read state machine to read data from the NAND flash. 0: disabled 1: enabled **Note**: read_data_en and write_data_en cannot be 1 at the same time.
[0]	RW	read_status_en	When this bit is 1, enable the command 0x70 that is used to sent the read status to the NAND flash, and then read status data from the NAND flash. The returned data is written to the NFC_STATUS field of the NANDC status register instead of the internal buffer. When erasing and programming the NAND flash, you need to read results to check whether operations are successful. If this bit is enabled, the operations such as programming and erasing can be complete at a time through the CPU and the data indicating whether the operations are successful is returned by the NAND flash. In this way, CPU intervention is reduced. **Note**: When read_data_en is 1, this bit is invalid.

NFC_STATUS

NFC_STATUS is the status register.

Offset Address	Register Name	Total Reset Value
0x20	NFC_STATUS	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ecc_num																reserved				nf_status								reserved				ready	nfc_ready
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	RO	ecc_num	When ecc_type is 10 or 11, it indicates the number of error bits for each sector (including the extended data of six bytes if ext_dta_ecc_en is 1). bit[3:0]: number of error bits for sector_0 bit[7:4]: number of error bits for sector_1 bit[11:8]: number of error bits for sector_2 bit[15:12]: number of error bits for sector_3 When ecc_type is 01, it indicates the decoding result of each sector and ext_data. bit[1:0]: decoding result of sector_0 bit[3:2]: decoding result of sector_1 bit[5:4]: decoding result of sector_2 bit[7:6]: decoding result of sector_3 bit[9:8]: decoding result of ext_data_0 bit[11:10]: decoding result of ext_data_1 bit[13:12]: decoding result of ext_data_2 bit[15:14]: decoding result of ext_data_3 The decoding result is defined as follows: 00: no error 01: 1-bit error 10: check code error but no data error 11: uncorrectable error (errors of two bits or more)
[15:13]	-	reserved	Reserved.
[12:5]	RO	nf_status	Status data when the NAND flash is read back. These bits are valid only when read_status of NFC_OP is 1 and nfc_ready of NFC_STATUS is 1.
[4:2]	-	reserved	Reserved.
[1]	RO	ready	Read_busy signal status of the NAND flash.

[0]	RO	nfc_ready	Read_busy signal status of the NANDC. 0: The NANDC is being used. 1: The operation is complete and the next command can be received. When the NANDC is enabled by writing NFC_OP, this bit is cleared automatically.

NFC_INTEN

NFC_INTEN is the interrupt enable register.

Offset Address	Register Name	Total Reset Value
0x24	NFC_INTEN	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																							wr_buf_err_int_en	wr_buf_busy_int_en	err_invalid	err_valid	reserved			cs1_done_en	cs0_done_en	op_done_en
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:9]	-	reserved	Reserved.
[8]	RW	wr_buf_err_int_en	Error interrupt enable bit for writing the lock address. 0: disabled 1: enabled
[7]	RW	wr_buf_busy_int_en	Error interrupt enable bit for reading and writing the NANDC buffer through the CPU when the NANDC reads/writes data from/to the NAND flash. 0: disabled 1: enabled
[6]	RW	err_invalid	Interrupt enable when uncorrectable errors occur.
[5]	RW	err_valid	Interrupt enable when correctable errors occur.
[4:3]	-	reserved	Reserved.
[2]	RW	cs1_done_en	Interrupt enable when the ready/busy signal changes from low to high. 0: disabled 1: enabled

[1]	RW	cs0_done_en	Interrupt enable when the ready/busy signal changes from low to high. 0: disabled 1: enabled
[0]	RW	op_done_en	Interrupt enable when the current operation of the NANDC is complete. 0: disabled 1: enabled

NFC_INTS

NFC_INTS is the interrupt status register.

Offset Address	Register Name	Total Reset Value
0x28	NFC_INTS	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																							wr_buf_err_int	wr_buf_busy_int	err_invalid	err_vavid	reserved		cs1_done	cs0_done	op_done
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:9]	-	reserved	Reserved.
[8]	RO	wr_buf_err_int	Interrupt status when the lock address is written. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	wr_buf_busy_int	Interrupt status when the CPU reads and writes the NANDC buffer in the process of reading/writing data from/to the NAND flash by the NANDC. 0: No interrupt is generated. 1: An interrupt is generated.

[6]	RO	err_invalid	Interrupt status when uncorrectable errors occur. 0: No interrupt is generated. 1: An interrupt is generated. In 1-bit ECC mode, if two or more error bits occur in the checked 512-byte data, an interrupt is generated. In 4-bit ECC mode, if five or more error bits occur in the checked 512-byte data, an interrupt is generated. In 8-bit ECC mode, if eight or more error bits occur in the checked 512-byte data, an interrupt is generated.
[5]	RO	err_vavid	Interrupt status when correctable errors occur. 0: No interrupt is generated. 1: An interrupt is generated. In 1-bit ECC mode, if 1-bit error occurs in the checked 512-byte data, an interrupt is generated. In 4-bit ECC mode, if 1-bit to 4-bit errors occur in the checked 512-byte data, an interrupt is generated. In 8-bit ECC mode, if 1-bit to 8-bit errors occur in the checked 512-byte data, an interrupt is generated.
[4:3]	-	reserved	Reserved.
[2]	RO	cs1_done	Interrupt status when the ready/busy signal changes from low to high. 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	cs0_done	Interrupt status when the ready/busy signal changes from low to high. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	op_done	Interrupt status when the current operation of the NANDC is complete. 0: No interrupt is generated. 1: An interrupt is generated. After NFC_OP is written, this bit is automatically cleared.

NFC_INTCLR

NFC_INTCLR is the interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x2C	NFC_INTCLR	-

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

Name	reserved	wr_buf_err_int_clr	wr_buf_busy_int_clr	r_5bit_err_clr	r_4bit_err_clr	reserved	cs1_done_clr	cs0_done_clr	op_done_clr
Reset	? ?	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:9]	-	reserved	Reserved.
[8]	WO	wr_buf_err_int_clr	wr_buf_err_int interrupt clear. 0: do not clear 1: clear
[7]	WO	wr_buf_busy_int_clr	wr_buf_busy_int interrupt clear. 0: do not clear 1: clear
[6]	WO	r_5bit_err_clr	r_5bit_err interrupt clear. 0: do not clear 1: clear
[5]	WO	r_4bit_err_clr	r_4bit_err interrupt clear. 0: do not clear 1: clear
[4:3]	-	reserved	Reserved.
[2]	WO	cs1_done_clr	cs1_done interrupt clear. 0: do not clear 1: clear
[1]	WO	cs0_done_clr	cs0_done interrupt clear. 0: do not clear 1: clear
[0]	WO	op_done_clr	op_done interrupt clear. 0: do not clear 1: clear

NFC_LOCK

NFC_LOCK is the lock address configuration register.

Offset Address	Register Name	Total Reset Value
0x30	NFC_LOCK	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																												lock_excmd_en	lock_en	global_lock_en	lock_down
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0

Bits	Access	Name	Description
[31:4]	-	reserved	Reserved.
[3]	RW	lock_excmd_en	Protection address write-protection enable according to the extended write command (new commands may be added for new memories). 0: disabled 1: enabled
[2]	RW	lock_en	Flash lock enable. When this control bit is 1, if the erased or programmed address is between the latch header address and the latch end address, the erasing and programming operations are invalid. 0: disabled 1: enabled
[1]	RW	global_lock_en	Flash global lock enable. When this bit is 1, erasing or programming operation is forbidden for the NAND flash. 0: disabled 1: enabled
[0]	RW	lock_down	NAND flash lock mode. 0: lock mode 1: lock_down mode. After the value 1 is written, this bit cannot be written any more. In addition, this bit can be cleared only when hardware is reset.

NFC_LOCK_SA0

NFC_LOCK_SA0 is the configuration register of lock start address 0.

Offset Address	Register Name	Total Reset Value
0x34	NFC_LOCK_SA0	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												flash_lock_cs				flash_lock_addr0															
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:19]	RW	flash_lock_cs	NAND flash latch CS. 00: CS 0 01: CS 1 Others: reserved
[18:0]	RW	flash_lock_addr0	Latch header address 0. The least significant bit (LSB) maps to the fifth row address of the NAND flash.

NFC_LOCK_SA1

NFC_LOCK_SA1 is the configuration register of lock start address 1.

Offset Address	Register Name	Total Reset Value
0x38	NFC_LOCK_SA1	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												flash_lock_cs				flash_lock_addr1															
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:19]	RW	flash_lock_cs	NAND flash latch CS. 00: CS 0 01: CS 1 Others: reserved

[18:0]	RW	flash_lock_addr1	Latch header address 1. The LSB maps to the fifth row address of the NAND flash.

NFC_LOCK_SA2

NFC_LOCK_SA2 is the configuration register of lock start address 2.

Offset Address	Register Name	Total Reset Value
0x3C	NFC_LOCK_SA2	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												flash_lock_cs				flash_lock_addr2															
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:19]	RW	flash_lock_cs	NAND flash latch CS. 00: CS 0 01: CS 1 Others: reserved
[18:0]	RW	flash_lock_addr2	Latch header address 2. The LSB maps to the fifth row address of the NAND flash.

NFC_LOCK_SA3

NFC_LOCK_SA3 is the configuration register of lock start address 3.

Offset Address	Register Name	Total Reset Value
0x40	NFC_LOCK_SA3	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												flash_lock_cs				flash_lock_addr3															

Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:19]	RW	flash_lock_cs	NAND flash latch CS. 00: CS 0 01: CS 1 Others: reserved
[18:0]	RW	flash_lock_addr3	Latch header address 3. The LSB maps to the fifth row address of the NAND flash.

NFC_LOCK_EA0

NFC_LOCK_EA0 is the configuration register of lock end address 0.

Offset Address	Register Name	Total Reset Value
0x44	NFC_LOCK_EA0	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												flash_lock_cs				flash_lock_eaddr0															
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:19]	RW	flash_lock_cs	NAND flash latch CS. 00: CS 0 01: CS 1 Others: reserved
[18:0]	RW	flash_lock_eaddr0	Latch end address 0. The LSB maps to the fifth row address of the NAND flash.

NFC_LOCK_EA1

NFC_LOCK_EA1 is the configuration register of lock end address 1.

Offset Address	Register Name	Total Reset Value
0x48	NFC_LOCK_EA1	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												flash_lock_cs				flash_lock_eaddr1															
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:19]	RW	flash_lock_cs	NAND flash latch CS. 00: CS 0 01: CS 1 Others: reserved
[18:0]	RW	flash_lock_eaddr1	Latch end address 1. The LSB maps to the fifth row address of the NAND flash.

NFC_LOCK_EA2

NFC_LOCK_EA2 is the configuration register of lock end address 2.

Offset Address	Register Name	Total Reset Value
0x4C	NFC_LOCK_EA2	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												flash_lock_cs				flash_lock_eaddr2															
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:19]	RW	flash_lock_cs	NAND flash latch CS. 00: CS 0 01: CS 1 Others: reserved

[18:0]	RW	flash_lock_eaddr2	Latch end address 2. The LSB maps to the fifth row address of the NAND flash.

NFC_LOCK_EA3

NFC_LOCK_EA3 is the configuration register of lock end address 3.

Offset Address	Register Name	Total Reset Value
0x50	NFC_LOCK_EA3	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												flash_lock_cs	flash_lock_eaddr3																			
Reset	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:19]	RW	flash_lock_cs	NAND flash latch CS. 00: CS 0 01: CS 1 Others: reserved
[18:0]	RW	flash_lock_eaddr3	Latch end address 3. The LSB maps to the fifth row address of the NAND flash.

NFC_EXPCMD

NFC_EXPCMD is the extended page command register.

Offset Address	Register Name	Total Reset Value
0x54	NFC_EXPCMD	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ex_pcmd3								ex_pcmd2								ex_pcmd1								ex_pcmd0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RW	ex_pcmd3	Extended page write command 3 of the NAND flash.
[23:16]	RW	ex_pcmd2	Extended page write command 2 of the NAND flash.

[15:8]	RW	ex_pcmd1	Extended page write command 1 of the NAND flash.
[7:0]	RW	ex_pcmd0	Extended page write command 0 of the NAND flash.

NFC_EXBCMD

NFC_EXPCMD is the extended block command register.

Offset Address	Register Name	Total Reset Value
0x58	NFC_EXBCMD	-

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																ex_bcmd1								ex_bcmd0							
Reset	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:8]	RW	ex_bcmd1	Extended block write command 1 of the NAND flash.
[7:0]	RW	ex_bcmd0	Extended block write command 0 of the NAND flash.

NFC_ECC_TEST

NFC_ECC_TEST is the ECC test register.

Offset Address	Register Name	Total Reset Value
0x5C	NFC_ECC_TEST	0x0020_F001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	version																empty				reserved										ecc_mask	dec_only	enc_only
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1

Bits	Access	Name	Description
[31:16]	RO	version	NANDC version number.
[15:12]	RO	empty	Empty flag of the internal first-in-first-out (FIFO) of the NANDC. bit[15]: empty flag of the ECC8 asynchronous FIFO bit[14]: empty flag of the ECC synchronous FIFO bit[13]: empty flag when the FIFO of the NAND flash is written bit[12]: empty flag when the FIFO of the NAND flash is read
[11:3]	-	reserved	Reserved.

[2]	RW	ecc_mask	ECC function mask. When this bit is 0, ECC check and correction are performed according to the value of ecc_type. When this bit is 1, ECC check and correction are forbidden. The structure of the data read from or written to the NAND flash is still converted based on the format of ecc_type.
[1]	RW	dec_only	Decoding enable only. When the value 1 is written to this bit, ECC decoding is enabled, but the NAND flash is not read or written. When this bit is read, the value 0 is returned.
[0]	RW	enc_only	Encoding enable only. When the value 1 is written to this bit, ECC encoding is enabled, but the NAND flash is not read or written. When this bit is read, the return value 1 indicates that ECC encoding and decoding are complete; the value 0 indicates that ECC encoding and decoding are being performed.

Contents

Figures

Tables

5 ETH

5.1 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, thus limiting the traffic of the CPU port to protect the CPU against heavy traffic.

5.2 Function Description

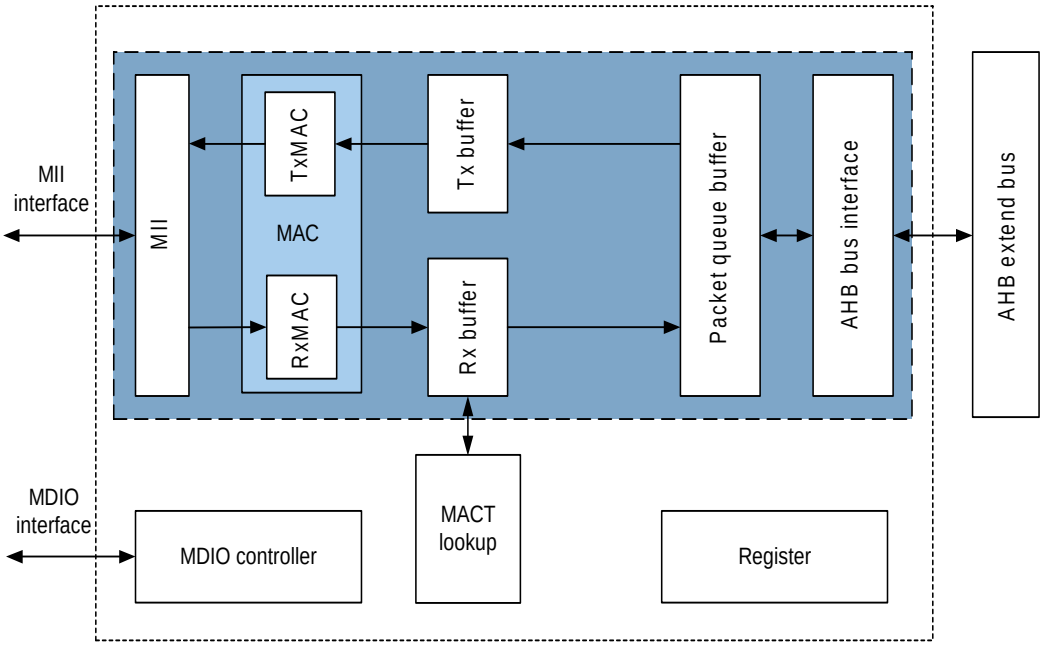
The ETH module has the following features:

- Supports one ETH module interface.
- Supports the rate of 10 Mbit/s or 100 Mbit/s.
- Supports full-duplex or half-duplex operating mode.
- Supports the MII interface.
- 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 the loopback to internal and loopback to external in full-duplex mode.
- Supports auto-adaption to automatically query the working state of the physical layer entity sublayer (PHY) chip.
- Provides the management data input/output (MDIO) interfaces with configurable frequency.
- Provides 64 frame management queues for both data receive and data transmit.
- Provides traffic limit to prevent the CPU against traffic attack.
- Supports the count of received frames and transmitted frames.

- Provides a receive buffer of 512 bytes and a transmit buffer of 2048 bytes.
- Supports eight configurable DMAC address filter tables.
- Controls whether to forward or discard broadcast frames, multicast frames, and unicast frames.

Figure 5-1 shows the logic block diagram of the ETH module.

Figure 5-1 Logic block diagram of the ETH module



5.3 Signal Description

Table 5-1 and Table 5-2 list signals of the ETH interface.

Table 5-1 MDIO interface signals

Signal	Direction	Description	Pin
MDCK	O	Clock output of MDIO interface	MDCK
MDIO	I/O	Input/output signal of the MDIO interface	MDIO

Table 5-2 MII interface signals

Signal	Direction	Description	Pin
MII_TXCK	I	MII transmit data clock	ETXCK
MII_TXD[3:0]	O	MII transmit data	ETXD3–ETXD0
MII_TXEN	O	MII transmit data valid	ETXEN
MII_RXCK	I	MII receive data clock	ERXCK
MII_RXD[3:0]	I	MII receive data	ERXD3–ERXD0
MII_RXDV	I	MII receive data valid	ERXDV
MII_CRS	I	MII carrier valid	ECRS
MII_COL	I	MII collision	ECOL

5.4 Operating Mode

5.4.1 Timings of Interfaces

Timings of the MII Interface

The Hi3515 provides standard MII interfaces that comply with the MII interface timing standard. These interfaces are used to connect to PHY chip.

Figure 5-2 shows the 100 Mbit/s receive timing of the MII interface.

Figure 5-2 100 Mbit/s receive timing of the MII interface

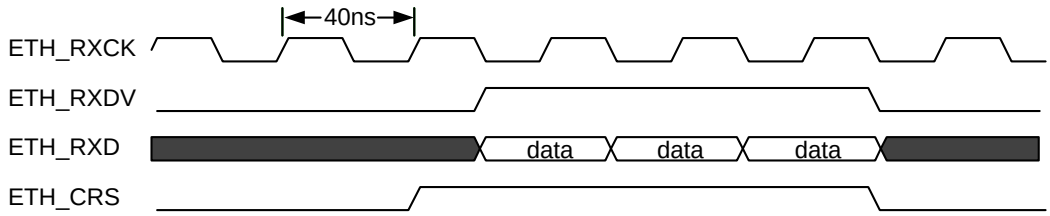


Figure 5-3 shows the 100 Mbit/s transmit timing of the MII interface.

Figure 5-3 100 Mbit/s transmit timing of the MII interface

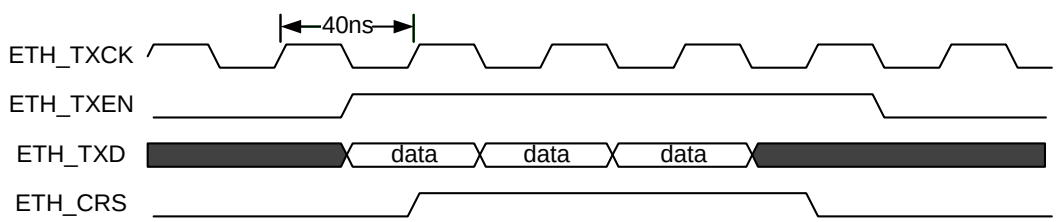


Figure 5-4 shows the 10 Mbit/s receive timing of the MII interface.

Figure 5-4 10 Mbit/s receive timing of the MII interface

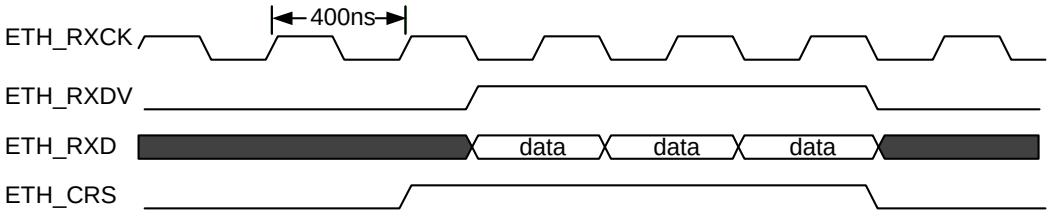


Figure 5-5 shows the 10 Mbit/s transmit timing of the MII interface.

Figure 5-5 10 Mbit/s transmit timing of the MII interface

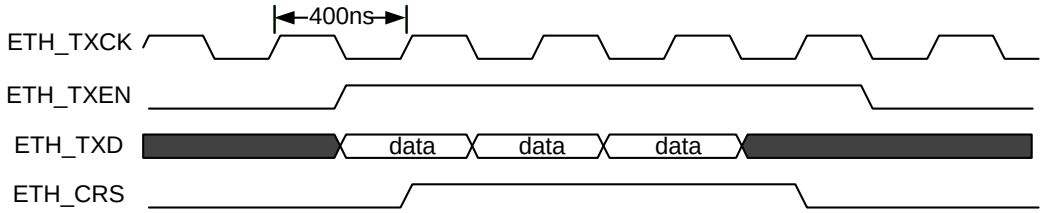


Figure 5-6 shows the receive timing parameters of the MII interface.

Figure 5-6 Receive timing parameters of the MII interface

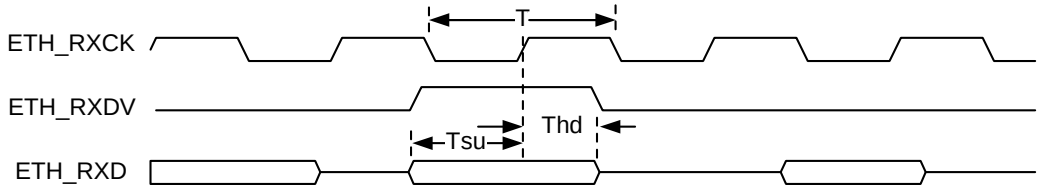


Figure 5-7 shows the transmit timing parameters of the MII interface.

Figure 5-7 Transmit timing parameters of the MII interface

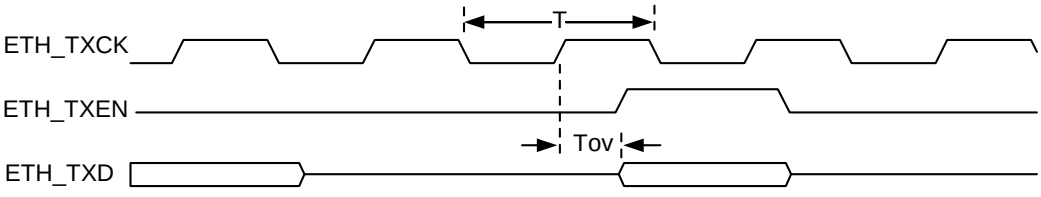


Table 5-3 lists the timing parameters of the MII interface.

Table 5-3 Timing parameters of the MII interface

Parameter	Symbol	Signal	Min	Max	Unit
MII clock cycle	T	ETH_RXCK, ETH_TXCK	400(10Mbit/s)	400	ns
			40(10Mbit/s)	40	ns
MII signal setup time	Tsu(RX)	ETH_RXER, ETH_RXDV, ETH_RXD[3:0]	6	–	ns
MII signal hold time	Thd(RX)	ETH_RXER, ETH_RXDV, ETH_RXD[3:0]	2	–	ns
MII output signal delay	Tov(MIITX)	ETH_TXD[1:0], ETH_TXEN	2	8	ns

Timings of the MDIO Interface

The MDIO interface implements the read/write control for the PHY chip. During software operation, the MDIO interface writes data, the address of the PHY chip, the address of the register, and related control information to the MDIO_RWCTRL register. If MDIO_RWCTRL[finish] is 1, it indicates that hardware has finished the read/write operation on the PHY chip. Hardware converts addresses, data, and control information to the timings of the MDIO interface. The read data is saved to MDIO_RO_DATA. After the CPU queried that MDIO_RWCTRL[finish] is 1, it reads data from MDIO_RO_DATA.

Through the MDIO interface, the Hi3515 can automatically acquire the working status of the PHY chip. If you want to enable the ETH module to work in auto-adaption mode, you need to configure the information such as address about the related status registers in the UD_MDIO_PHYADDR and UD_MDIO_ANEG_CTRL registers. The ETH module automatically reads the status information from the related register of the specified PHY chip through the MDIO interface and then stores the status information to the UD_MDIO_RO_STAT register. For details about the address of the specific status register, see the data sheet related to the PHY chip.

Figure 5-8 shows the read timing of the MDIO interface.

Figure 5-8 Read timing of the MDIO interface

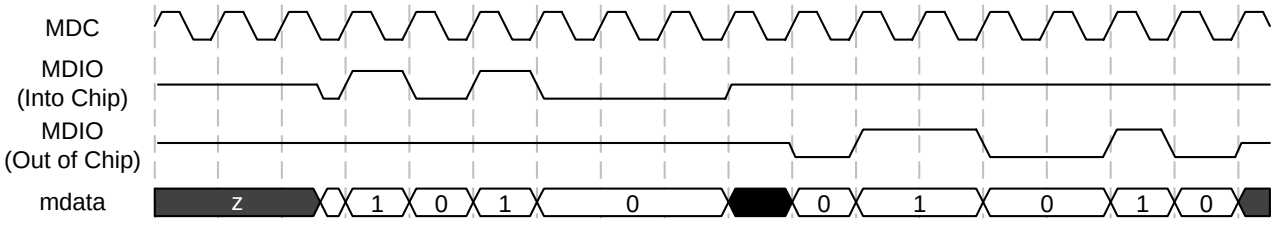


Figure 5-9 shows the write timing of the MDIO interface.

Figure 5-9 Write timing of the MDIO interface

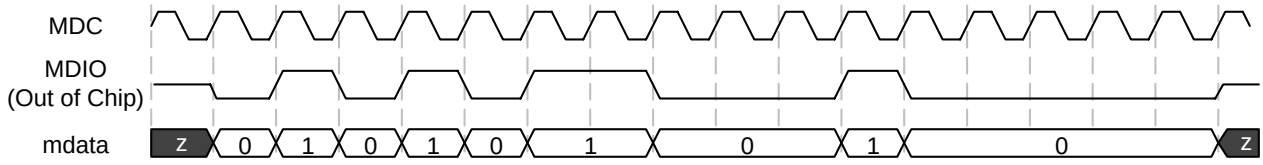


Figure 5-10 shows the timing parameters of the MDIO interface.

Figure 5-10 Receive timing parameters of the MDIO interface

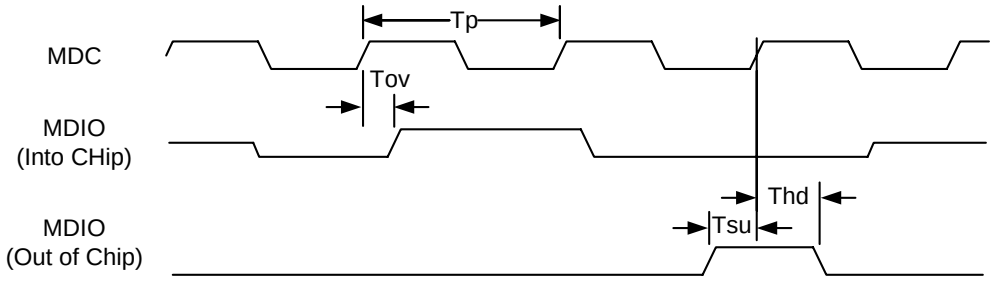


Table 5-4 lists the timing parameters of the MDIO interface.

Table 5-4 Timing parameters of the MDIO interface

Parameter	Symbol	Signal	Min	Max	Unit
MDIO data receive delay time	Tov	MDIO	166	20833	ns
MDIO clock cycle	Tp	MDC	333	41667	ns
MDIO data transmit setup time	Tsu	MDIO	10	–	ns
MDIO data transmit hold time	Thd	MDIO	10	–	ns

The MDC clock cycle Tp can be changed by adjusting the MDC frequency (MDIO_RWCTRL[frq_dv]). To be specific, you can divide the frequency 150 MHz of the ETH module working clock by 100, 50, or other values. Tov is related to the clock period Tp of the MDC and it is about Tmdc/2.

5.4.2 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 receive queue depths and the size of each buffer is 2 KB. Then, software writes the header addresses of the buffers to the frame receive queue one by one. The times of the write operation is equal to the configured receive 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 receive 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 receive queue. Otherwise, the available depth of receive queues equals to the number of buffers assigned to the CPU rather than the value configured by the CPU.

Table 5-5 describes the data structure of the frame descriptor received by the CPU.

Table 5-5 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 intput 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 receive queue can be obtained by querying UD_GLB_ADDRQ_STAT.

The CPU can receive a frame in interrupt or query mode.

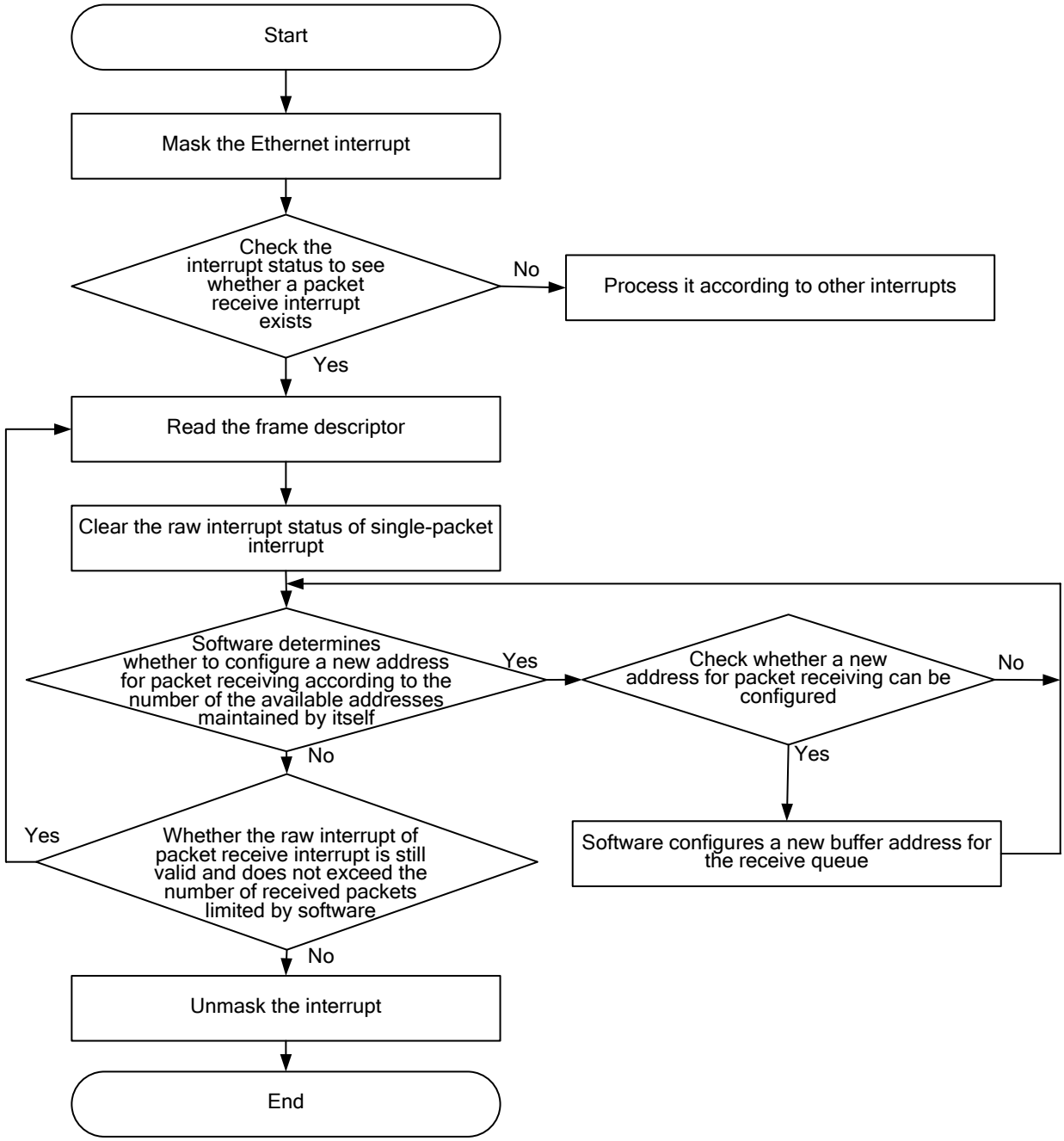
1. Receiving a frame in interrupt mode

   When the CPU enables the frame receive interrupt, depending on the frames to be received, hardware generates frame receive 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 5-11 shows the process of receiving a frame in interrupt mode.

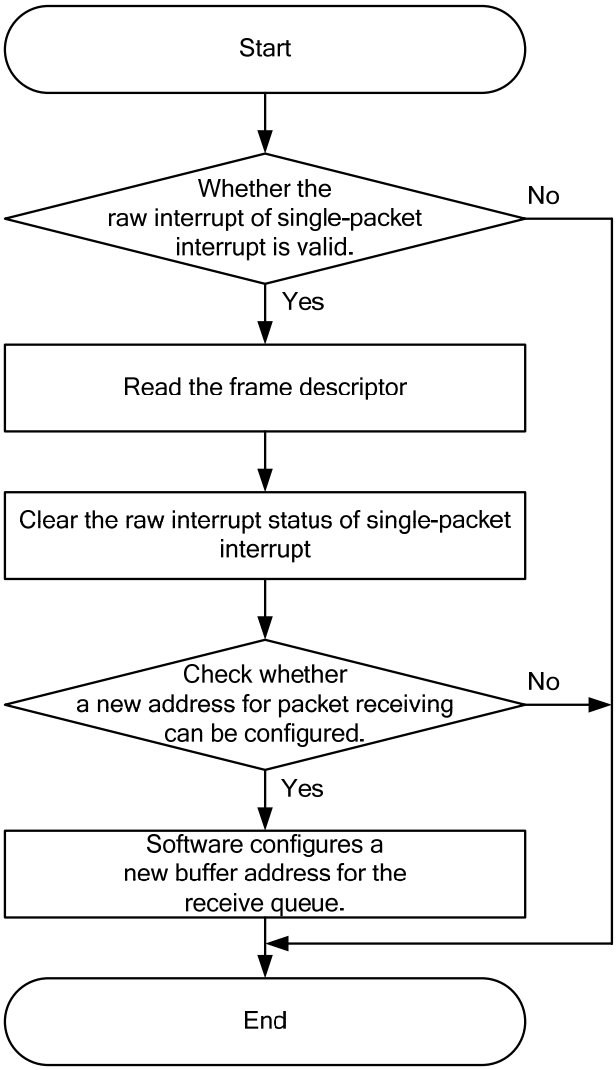
Figure 5-11 Process of receiving a frame in interrupt mode



2. Receiving a frame in query mode

   In this mode, the CPU does not enable the frame receive 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 5-12 shows the process of receiving a frame in query mode.

Figure 5-12 Process of receiving a frame in query mode



5.4.3 Process of Transmitting a Frame

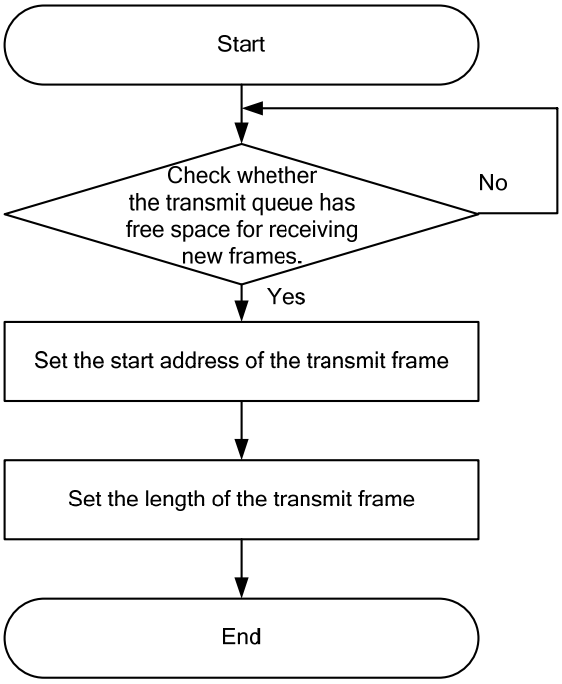
When a frame is transmitted, the CPU checks whether the current queue has any available space. If the space is sufficient, the CPU writes the header address of the buffer and then the length of the transmit frame to the frame descriptor of the transmit queue. The frame length trigger hardware for writing the transmit frame writes the header address and frame length of the transmit frame to the transmit 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 5-13 shows the process of transmitting a frame by the CPU.

Figure 5-13 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 5-6 describes the data structure of the frame descriptor transmitted by the CPU.

Table 5-6 Data structure of the frame descriptor transmitted by the CPU

Bits	Name	Description
[42:11]	start_addr_eq	Header address of a frame.
[10:0]	fm_len	Frame length in the unit of byte.

Note: The frames whose fm_len 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 transmit 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 transmit queue of the ETH module and allows the interrupt to be notified to the CPU. If the transmit 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 transmit queue is full, software enables the interrupt. After the transmit queue is empty, an interrupt occurs to notify software of transmitting 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 transmit frames. If the count is less than the configured depth of transmit queue, it notifies the ETH module of the to-be transmitted frame directly. At the same time, it creates a corresponding transmit frame index table whose content is the header address of the frame that is written to the transmit queue of the Ethernet MAC. After transmitting a frame, the ETH module notifies the CPU of releasing the corresponding transmit buffer through the address of the transmit queue. After that, the CPU queries the corresponding transmit buffer through the address of the transmit queue and releases the buffer.

5.4.4 Interrupt Management

Interrupt Status Register

This register indicates the generated interrupt type. For details, see GLB_IRQ_STAT in section 5.6.3 "Description of the Global Control Registers."

Interrupt Enable Register

This register controls whether to generate the related interrupts. For details, see GLB_IRQ_ENA in section 5.6.3 "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 5.6.3 "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.

5.4.5 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.

5.4.6 MAC Forwarding

For the receive frames whose destination MAC address is the same as the local MAC address, the logic directly forwards the frame to the CPU port.

For the receive frames whose destination MAC address is not the same as the local MAC address, the logic performs the following operations:

Step 1 Determine whether the receive frame is a broadcast frame. If the receive frame is a broadcast frame, perform the operations related to the broadcast frame. Otherwise, go to step 2.

Step 2 Determine whether the function of querying the MAC forward control table (controlled by GLB_MACTCTRL bit[15] and GLB_MACTCTRL bit[7]) is enabled. If the function is enabled, go to step 3. Otherwise, go to step 4.

Step 3 Determine whether the receive frame matches the MAC forward control table. If the receive frame matches the MAC forward control table, forward the receive frame according to the entry configuration. Otherwise, go to step 4.

Step 4 Determine whether the receive frame is a multicast frame or unicast frame. If the receive frame is a multicast frame or unicast frame, perform the operations related to multicast frames or unicast frames. Otherwise, discard the receive frame.

----End

The MAC forward control table can be configured through the addresses (GLB_MAC0_L32 to GLB_MAC7_H16). The method of configuring the MAC forward control table is as follows:

- The local MAC address is configured through the addresses GLB_HOSTMAC_L32 and GLB_HOSTMAC_H16.
- The MAC0 entry is configured through the addresses GLB_MAC0_L32 and GLB_MAC0_H16.
- The MAC1 entry is configured through the addresses GLB_MAC1_L32 and GLB_MAC1_H16.
- The MAC2 entry is configured through the addresses GLB_MAC2_L32 and GLB_MAC2_H16.
- The MAC3 entry is configured through the addresses GLB_MAC3_L32 and GLB_MAC3_H16.
- The MAC4 entry is configured through the addresses GLB_MAC4_L32 and GLB_MAC4_H16.
- The MAC5 entry is configured through the addresses GLB_MAC5_L32 and GLB_MAC5_H16.
- The MAC6 entry is configured through the addresses GLB_MAC6_L32 and GLB_MAC6_H16.
- The MAC7 entry is configured through the addresses GLB_MAC7_L32 and GLB_MAC7_H16.

5.4.7 Typical Application

Pin Multiplexing Configuration

Table 5-7 describes the ETH pin multiplexing.

Table 5-7 ETH pin multiplexing

Signal	Enable Configuration	Multiplexing Description
ECOL	reg46	Multiplexed with GPIO1_1
ECRS	reg47	Multiplexed with GPIO1_2

Clock Gating

NOTE

When the ETH module is not used, its clocks can be disabled to reduce the power consumption.

To disable the ETH clocks, the following steps take place:

Step 1 Disable the link status of the ETH interface so that the ETH module cannot transmit or receive packets.

Step 2 Clear the receive queues of ETH interface so that the ETH module cannot report the packet receive interrupt.

Step 3 Software delivers a logic command to reset the ETH module and holds the reset status.

Step 4 Set SC_PERDIS[ethclkdis] to 1 to disable the ETH clocks.

----End

To enable the ETH clocks, the following steps take place:

Step 1 Hold the reset status. Set SC_PEREN[ethclken] to 1 to enable the ETH clocks.

Step 2 Set SC_PERCTRL8[eth_srst] to 1 to clear the reset status.

Step 3 Enable the link status of the ETH interface to ensure that the ETH module works properly.

----End

Soft Reset

To perform global soft reset on the ETH module, the following steps take place:

Step 1 Disable the link status of the ETH interface and the packet receive interrupt so that software cannot receive or transmit packets.

Step 2 After processing the current received and transmitted packets on the ETH interface, software clears the receive and transmit queues and keeps the queue length the same as the value before soft reset. That is, the count values of the related pointers and queues return to 0.

Step 3 Set SC_PERCTRL8[eth_srst] to 0 to deliver the soft reset command for the ETH module.

Step 4 Set SC_PERCTRL8[eth_srst] to 1 to clear the soft reset on the ETH module.

Step 5 If packets need to be transmitted and received again, software also needs to initialize the receive and transmit queues of the ETH interface.

Step 6 Enable the link status of the ETH interface to ensure that the ETH module works properly.

----End

Initialization

To initialize the ETH interface, the following steps take place:

Step 1 Configure the mode for obtaining the interface status.

The status of the ETH interface can be that of the PHY chip by means of auto-adaption or can be configued by the software. During initialization, select the mode for obtaining the interface status by configuring UD_MAC_PORTSEL[stat_ctrl]:

- If UD_MAC_PORTSEL[stat_ctrl] is set to 1, it indicates that the status of the ETH interface is configured by the software. In this case, go to Step 2.
- If UD_MAC_PORTSEL[stat_ctrl] is set to 0, it indicates that the status of the ETH interface is that of the PHY chip by means of auto-adaption. In this case, go to Step 3.

During reset, software configures the working status of the ETH interface.

Step 2 Configure the working status of the PHY chip.

- If UD_MAC_PORTSEL is set to 1, software needs to configure the rate, connection status, and duplex status in UD_MAC_PORTSET according to the actual application environment, and configures the information to the related registers of the PHY chip.
- The ETH module provides a MDIO interface to implement the read/write control for the PHY chip. During software operation, the MDIO interface writes the address of the PHY chip, the address of the register, and related control information to the MDIO_RWCTRL register. When MDIO_RWCTRL[finish] is 1, it indicates the read/write operation on the PHY chip has been finished by hardware. For details about configuration information, see the data sheet related to the PHY chip.

After the configuration is complete, go to Step 4.

Step 3 Configure the working status in auto-adaption mode.

If UD_MAC_PORTSEL[stat_ctrl] is set to 0, you must specify the rate of the PHY chip, duplex mode, address of the connection register, and offset addresses of the registers of such status bits. The information is configured through UD_MDIO_ANEG_CTRL.

Step 4 Set the depth of receive and transmit queues.

Set the receive queue depth and transmit queue depth in the register UD_GLB_QLEN_SET:

- The receive queue depth indicates the maximum number of buffered frames when data is received.
- The transmit queue depth indicates the maximum number of buffered frames when data is transmitted.

The receive and transmit queues share 64 management spaces, so the sum of the receive queue depth and the transmit queue depth cannot exceed 64. Additionally, either the receive queue depth or the transmit queue depth must be more than or equal to 1. If the sum exceeds 64, the depth of the receive queue remains unchanged and the depth of the transmit queue is changed to 64 minus the depth of the receive queue.

Software can set the multi-packet interrupt configuration register. By configuring the register UD_GLB_IRQN_SET[int_frm_cnt], software controls how many packets need to be received before a multi-packet interrupt is reported. In addition, software can set the aging time register UD_GLB_IRQN_SET[int_timer].

Step 5 Initialize the buffer of the receive frame queue.

After the reset, software needs to apply for a certain number of buffers that is equal to the configured depth of receive queues. The size of each buffer is 2 KB. Then, software writes the header addresses of the buffers to the receive queue. The number of times of the write operation must be equal to the configured depth of receive queues.

Step 6 Execute the soft reset on the ETH module.

Through the soft reset, the logic circuits and frame management queues inside the ETH module are reset, so that the ETH module returns to the initial status. However, the registers inside the ETH module keep the original values. After clearing the reset, software re-applies for a packet receive buffer and initializes the receive queue. Otherwise, the ETH module cannot receive network packets.

NOTE

After the soft reset of the ETH module, the registers configured by software remain unchanged. For details about these registers, see the register description.

----End

Process of Receiving a Frame in Interrupt Mode

To receive a frame in interrupt mode, the following steps take place:

Step 1 Mask the Ethernet interrupt after entering the interrupt handling program.

Step 2 View the interrupt status bit GLB_IRQ_STAT[int_rx_up] to check whether a frame receive interrupt occurs. If a frame receive interrupt occurs, go to Step 2. Otherwise, continue to process other Ethernet interrupts.

Step 3 Read the frame descriptor UD_GLB_IQFRM_DES. Read the frame data of the related length (fd_in_len) according to the header address of the frame that corresponds to fd_in_addr.

Step 4 Return the raw interrupt signal bit GLB_IRQ_RAW[iraw_rx_up] of the single-packet interrupt to 0 and notify hardware of the completion of packet receiving.

Step 5 According to the number of the remaining available addresses maintained by itself, software determines whether to configure a new address for packet receiving. If a new address for packet receiving does not need to be configured, go to Step 7. Receiving a packet is complete.

Step 6 Read UD_GLB_QSTAT[cpu_addr_in_rdy] to check whether a new address for packet receiving can be configured. If a new address for packet receiving cannot be configured, return to Step 4.

Step 7 Software allocates a new buffer address to the receive queue through the register UD_GLB_IQ_ADDR. Return to Step 4.

Step 8 Read and check the raw interrupt signal bit GLB_IRQ_RAW[iraw_rx_up] of the single-packet interrupt. If the bit is valid and software can continue to receive packets (the upper limit of received packets is not exceeded), return to Step 2.

Step 9 Unmask the Ethernet interrupt.

----End

Process of Receiving a Frame in Query Mode

If the frame receive interrupt GLB_IRQ_ENA[ien_cpu_rx] is disabled, the CPU automatically queries the raw interrupt signal bit GLB_IRQ_RAW[iraw_rx_up] of the single-packet interrupt. If the bit is set to 1, it indicates that a frame needs to be received.

Receive a frame in query mode as follows:

Step 1 Read the raw interrupt signal bit GLB_IRQ_RAW[iraw_rx_up] of the single-packet interrupt. If this bit is invalid, the process ends.

Step 2 Read the frame descriptor UD_GLB_IQFRM_DES. Read the frame data of the related length (fd_in_len) according to the header address of the frame that corresponds to fd_in_addr.

Step 3 Write 1 to clear the raw interrupt signal bit GLB_IRQ_RAW[iraw_rx_up] of the single-packet interrupt and notify hardware of the completion of packet receiving.

Step 4 Read UD_GLB_QSTAT[cpu_addr_in_rdy] to check whether a new address for packet receiving can be configured. If a new address for packet receiving cannot be configured, the process ends.

Step 5 Software allocates a new buffer address to the receive queue through the register UD_GLB_IQ_ADDR.

----End

Process of Transmitting a Frame

Transmit a frame as follows:

Step 1 Read UD_GLB_ADDRQ_STAT[eq_in_rdy] and check whether the transmit queue of the ETH module has free space for receiving new transmit frames. If the transmit queue of the ETH module has no free space, continue to wait and query for a free space.

Step 2 Configure the header address UD_GLB_EQ_ADDR of the frame to be transmitted.

Step 3 Configure the length UD_GLB_EQFRM_LEN of the frame to be transmitted. The configuration for transmitting a frame is complete.

----End

5.5 Register Summary

MDIO Control Registers

Table 5-8 lists the MDIO control registers.

Table 5-8 Summary of the MDIO control registers (base address: 0x1009_0000)

Offset Address	Name	Description	Page
0x1100	MDIO_RWCTRL	MDIO command word register	5-22
0x1104	MDIO_RO_DATA	MDIO read data register	5-24

Offset Address	Name	Description	Page
0x0108	UD_MDIO_PHYADDR	PHY physical address register	5-24
0x010C	UD_MDIO_RO_STAT	PHY chip status register	5-24
0x0110	UD_MDIO_ANEG_CTRL	Offset address configuration register for the PHY chip status	5-25
0x0114	UD_MDIO_IRQENA	Scan mask register for MDIO status changes	5-26

MAC Control Registers

Table 5-9 lists the MAC control registers.

Table 5-9 Summary of the MAC controller registers (base address: 0x1009_0000)

Offset Address	Name	Description	Page
0x0200, 0x2200	UD_MAC_PORTSEL	Port working status control register	5-27
0x0204, 0x2204	UD_MAC_RO_STAT	Port status register	5-28
0x0208, 0x2208	UD_MAC_PORTSET	Port working status configuration register	5-28
0x020C, 0x220C	UD_MAC_STAT_CHANGE	Port status change indicator register for the MAC	5-29
0x0210, 0x2210	UD_MAC_SET	MAC function configuration register	5-30

Global Control Registers

Table 5-10 lists the Ethernet global control registers.

Table 5-10 Summary of the global control registers (base address: 0x1009_0000)

Offset Address	Name	Description	Page
0x1300	GLB_HOSTMAC_L32	Lower 32-bit register for the local MAC address.	5-32
0x1304	GLB_HOSTMAC_H16	Upper 16-bit register for the local MAC address.	5-32
0x1308	GLB_SOFT_RESET	Internal soft reset register.	5-32

Offset Address	Name	Description	Page
0x1310	GLB_FWCTRL	Forward control register.	5-33
0x1314	GLB_MACTCTRL	MAC filter table control register.	5-34
0x1318	GLB_ENDIAN_MOD	Endian control register.	5-35
0x1330	GLB_IRQ_STAT	Interrupt status register.	5-35
0x1334	GLB_IRQ_ENA	Interrupt enable register.	5-37
0x1338	GLB_IRQ_RAW	Raw interrupt register.	5-39
0x1400	GLB_MAC0_L32	MAC filter 0.	5-40
0x1404	GLB_MAC0_H16	MAC filter 0.	5-41
0x1408	GLB_MAC1_L32	MAC filter 1.	5-41
0x140C	GLB_MAC1_H16	MAC filter 1.	5-42
0x1410	GLB_MAC2_L32	MAC filter 2.	5-42
0x1414	GLB_MAC2_H16	MAC filter 2.	5-43
0x1418	GLB_MAC3_L32	MAC filter 3.	5-43
0x141C	GLB_MAC3_H16	MAC filter 3.	5-44
0x1420	GLB_MAC4_L32	MAC filter 4.	5-45
0x1424	GLB_MAC4_H16	MAC filter 4.	5-45
0x1428	GLB_MAC5_L32	MAC filter 5.	5-46
0x142C	GLB_MAC5_H16	MAC filter 5.	5-46
0x1430	GLB_MAC6_L32	MAC filter 6.	5-47
0x1434	GLB_MAC6_H16	MAC filter 6.	5-47
0x1438	GLB_MAC7_L32	MAC filter 7.	5-48
0x143C	GLB_MAC7_H16	MAC filter 7.	5-48
0x0340	UD_GLB_IRQN_SET	Multi-packet interrupt configuration register.	5-49
0x0344	UD_GLB_QLEN_SET	Queue length configuration register.	5-49
0x0348	UD_GLB_FC_LEVEL	Traffic control register.	5-50
0x034C	UD_GLB_CAUSE	Cause register for the CPU to which the packet is transmitted.	5-51
0x0350	UD_GLB_RXFRM_SADDR	Receive frame start address register.	5-51

Offset Address	Name	Description	Page
0x0354	UD_GLB_IQFRM_DES	Receive frame descriptor register.	5-52
0x0358	UD_GLB_IQ_ADDR	Receive frame header address register.	5-52
0x035C	UD_GLB_BFC_STAT	Counter for traffic control status of forward buffer and aging time of multi-packet interrupt.	5-53
0x0360	UD_GLB_EQ_ADDR	Transmit queue header address register.	5-53
0x0364	UD_GLB_EQFRM_LEN	Transmit queue frame length configuration register.	5-54
0x0368	UD_GLB_QSTAT	Queue status register.	5-55
0x036C	UD_GLB_ADDRQ_STAT	Address queue status register.	5-55
0x0370	UD_GLB_FC_TIMECTRL	Traffic control time configuration register.	5-56
0x0374	UD_GLB_FC_RXLIMIT	Traffic control limit configuration register.	5-57
0x0378	UD_GLB_FC_DROPCTRL	Packet drop control register for traffic control.	5-57

Statistics Counter Control Registers

Table 5-11 lists the statistics counter control registers.

Table 5-11 Summary of the statistics counter control registers (base address: 0x1009_0000)

Offset Address	Name	Description	Page
0x0584	UD_STS_PORTCNT	Port status counter.	5-58
0x05A0	UD_PORT2CPU_PKTS	Register for the total number of packets received by the CPU from the uplink or downlink port.	5-59
0x05A4	UD_CPU2IQ_ADDRCNT	Register for the count of configuring packet receiving address queue by the CPU.	5-59

Offset Address	Name	Description	Page
0x05A8	UD_RX_IRQCNT	Register for the count of reporting single-packet interrupt by the uplink or downlink port.	5-59
0x05AC	UD_CPU2EQ_PKTS	Register for the total number of packets transmitted by the CPU to the uplink or downlink port.	5-60

Statistics Result Registers

Table 5-12 lists the statistics result registers.

Table 5-12 Summary of the statistics result registers (base address: 0x1009_0000)

Offset Address	Name	Description	Page
0x0600	UD_RX_DVCNT	RXDV rising edge count register.	5-60
0x0604	UD_RX_OCTS	Register for the total number of received bytes.	5-61
0x0608	UD_RX_RIGHTOCTS	Register for the total number of bytes of received correct packets.	5-61
0x060C	UD_HOSTMAC_PKTS	Register for the number of packets matching the local MAC address.	5-62
0x0610	UD_RX_RIGHTPKTS	Register for the total number of packets received by the port.	5-62
0x0614	UD_RX_BROADPKTS	Register for the number of correct broadcast packets.	5-62
0x0618	UD_RX_MULTPKTS	Register for the number of correct multicast packets.	5-63
0x061C	UD_RX_UNIPKTS	Register for the number of correct unicast packets.	5-63
0x0620	UD_RX_ERRPKTS	Register for the total number of incorrect packets.	5-64
0x0624	UD_RX_CRCERR_PKTS	Register for the count of CRC errors.	5-64
0x0628	UD_RX_LENERR_PKTS	Register for the number of packets with invalid length.	5-64

Offset Address	Name	Description	Page
0x062C	UD_RX_OCRCERR_PKTS	Register for the number of packets with odd nibbles and CRC errors.	5-65
0x0630	UD_RX_PAUSE_PKTS	Register for the number of received pause packets.	5-65
0x0634	UD_RF_OVERCNT	Register for the count of RXFIFO overflow events.	5-65
0x0638	UD_FLUX_TOL_IPKTS	Register for the total number of received packets allowed by the traffic limit.	5-66
0x063C	UD_FLUX_TOL_DPKTS	Register for the total number of packets discarded due to traffic limit.	5-66
0x0640	UD_VN2OTH_PKTS	Register for the number of packets not forwarded to another port due to VLAN limit.	5-67
0x0644	UD_VN2CPU_PKTS	Register for the number of packets not forwarded to the CPU port due to VLAN limit.	5-67
0x0648	UD_MN2OTH_PKTS	Register for the number of packets not forwarded to another port due to MAC limit.	5-67
0x064C	UD_MN2CPU_PKTS	Register for the number of packets not forwarded to the CPU port due to MAC limit.	5-68
0x0780	UD_TX_PKTS	Register for the total number of packets transmitted successfully.	5-68
0x0784	UD_TX_BROADPKTS	Register for the number of broadcast packets transmitted successfully.	5-69
0x0788	UD_TX_MULTPKTS	Register for the number of multicast packets transmitted successfully.	5-69
0x078C	UD_TX_UNIPKTS	Register for the number of unicast packets transmitted successfully.	5-69
0x0790	UD_TX_OCTS	Register for the total number of transmitted bytes.	5-70

Offset Address	Name	Description	Page
0x0794	UD_TX_PAUSE_PKTS	Register for the number of transmitted pause frames.	5-70
0x0798	UD_TX_RETRYCNT	Register for the total count of retransmission.	5-70
0x079C	UD_TX_COLCNT	Register for the total count of collisions.	5-71
0x07A0	UD_TX_LC_PKTS	Register for the number of packets with late collision.	5-71
0x07A4	UD_TX_COLOK_PKTS	Register for the number of packets transmitted successfully with collisions.	5-71
0x07A8	UD_TX_RETRY15_PKTS	Register for the number of packets discarded due to more than 15 times of retransmission.	5-72
0x07AC	UD_TX_RETRYN_PKTS	Register for the number of packets with the count of collisions being equal to the threshold.	5-72

5.6 Register Description

5.6.1 Description of the MDIO Control Registers

MDIO_RWCTRL

MDIO_RWCTRL is the MDIO command word register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x1100	MDIO_RWCTRL	0x0000_8000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																finish	reserved	rw	phy_exaddr					frq_dv				phy_inaddr			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description

[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.
[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.
[14]	RO	reserved	Reserved.
[13]	RW	rw	PHY read or write access control. 0: Read operation. 1: Write operation.
[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 UD_MDIO_PHYADDR[phy0_addr] or UD_MDIO_PHYADDR[phy1_addr].
[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 clock as an example to describe the matching relations between frq_dv and MDC frequency. 000: The frequency of the working main clock is divided by 50 and thus the obtained frequency is 2 MHz. 001: The frequency of the working main clock is divided by 100 and thus the obtained frequency is 1 MHz. 010: The frequency of the working main clock is divided by 200 and thus the obtained frequency is 512 kHz. 011: The frequency of the working main clock is divided by 400 and thus the obtained frequency is 256 kHz. 100: The frequency of the working main clock is divided by 800 and thus the obtained frequency is 128 kHz. 101: The frequency of the working main clock is divided by 1600 and thus the obtained frequency is 64 kHz. 110: The frequency of the working main clock is divided by 3200 and thus the obtained frequency is 32 kHz. 111: The frequency of the working main clock is divided by 6400 and thus the obtained frequency is 16 kHz.
[4:0]	RW	phy_inaddr	Internal register address of the external PHY chip. This address is presented by a 5-bit binary number.

MDIO_RO_DATA

MDIO_RO_DATAMDIO is the read data register. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x1104	MDIO_RO_DATA	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[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.

UD_MDIO_PHYADDR

UD_MDIO_PHYADDR is the PHY physical address register. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0108	UD_MDIO_PHYADDR	0x0000_0001

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

Bits	Access	Name	Description
[31:5]	RO	reserved	Reserved.
[4:0]	RW	phy_addr	Physical address of the external PHY chip.

UD_MDIO_RO_STAT

UD_MDIO_RO_STAT is the PHY status register. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x010C	UD_MDIO_RO_STAT	0x0000_0000

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

Name	reserved	speed_mdio2mac	link_mdio2mac	duplex_mdio2mac
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			

Bits	Access	Name	Description
[31:3]	RO	reserved	Reserved.
[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 working mode. 1: 100 Mbit/s working mode.
[1]	RO	link_mdio2mac	Port link status obtained from the MDIO interface. 0: No link exists. 1: A link exists.
[0]	RO	duplex_mdio2mac	Port duplex working status obtained from the MDIO interface. 0: Half-duplex. 1: Full-duplex.

UD_MDIO_ANEG_CTRL

UD_MDIO_ANEG_CTRLPHY is the offset address configuration register for the PHY status. The register does not support soft reset.

NOTE

The PHY speed status is indicated by bit[14] of the register with the address of 17, internal_addr_speed is set to 0x11 and speed_index is set to 0xE. In this case, the ETH module reads the bit value as the current working speed mode of the PHY through the MDIO interface.

Offset Address	Register Name	Total Reset Value
0x0110	UD_MDIO_ANEG_CTRL	0x0463_1EA9

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	internal_addr_speed	internal_addr_link	internal_addr_duplex		speed_index	link_index	duplex_index
Reset	0 0 0 0	0 1 0 0	0 1 1 0	0 0 1 1	0 0 0 1	1 1 1 0	1 0 1 0	1 0 0 1

Bits	Access	Name	Description
[31:27]	RO	reserved	Reserved.

[26:22]	RW	internal_addr_speed	Address of the register in the PHY chip to store the status information (speed). The default value is set according to Intel 9785.
[21:17]	RW	internal_addr_link	Address of the register in the PHY chip to store the status information (link). The default value is set according to Intel 9785.
[16:12]	RW	internal_addr_duplex	Address of the register in the PHY chip to store the status information (duplex). The default value is set according to Intel 9785.
[11:8]	RW	speed_index	Offset address in the PHY status register that is used to store the speed information. The default value is set according to Intel 9785.
[7:4]	RW	link_index	Offset address in the PHY status register that is used to store the link information. The default value is set according to Intel 9785.
[3:0]	RW	duplex_index	Offset address in the PHY status register that is used to store the duplex information. The default value is set according to Intel 9785.

UD_MDIO_IRQENA

UD_MDIO_IRQENA is the scan mask register for MDIO status changes. The register does not support soft reset.

NOTE

- If the status information about the PHY chip connecting to the port cannot be scanned and obtained by configuring UD_MDIO_ANEG_CTRL, you can scan the PHY status register by using MDIO_RWCTRL to check whether the port status changes and generate an interrupt to notify software of processing the interrupt.
- link_partner status change refers to the change of any bit of link, speed, and duplex for the PHY status.

Offset Address	Register Name	Total Reset Value
0x0114	UD_MDIO_IRQENA	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																												link_partner_ch_mask		speed_ch_mask		link_ch_mask		duplex_ch_mask	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:4]	RO	reserved	Reserved.

[3]	RW	link_partner_ch_mask	Port link partner status scan change interrupt mask. 0: Mask. 1: Unmask.
[2]	RW	speed_ch_mask	Port speed mode scan change interrupt mask. 0: Mask. 1: Unmask.
[1]	RW	link_ch_mask	Port link mode scan change interrupt mask. 0: Mask. 1: Unmask.
[0]	RW	duplex_ch_mask	Port duplex mode scan change interrupt mask. 0: Mask. 1: Unmask.

5.6.2 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 the port working status control register. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0200	UD_MAC_PORTSEL	0x0000_0001

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

Bits	Access	Name	Description
[31:2	RO	reserved	Reserved.
[1]	RW	mii_rmii	Port interface mode selection. 0: MII interface. 1: RMII interface.
[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.

UD_MAC_RO_STAT

UD_MAC_RO_STAT is the port status register. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0204	UD_MAC_RO_STAT	0x0000_0000

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

Bits	Access	Name	Description
[31:3]	RO	reserved	Reserved.
[2]	RO	speed_stat	Port current speed mode. 0: 10 Mbit/s mode. 1: 100 Mbit/s mode.
[1]	RO	link_stat	Port current link status. 0: No link exists. 1: A link exists.
[0]	RO	duplex_stat	Port current duplex status. 0: Half-duplex. 1: Full-duplex.

UD_MAC_PORTSET

UD_MAC_PORTSET is the port working status configuration register. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0208	UD_MAC_PORTSET	0x0000_0000

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

Bits	Access	Name	Description

[31:3]	RO	reserved	Reserved.
[2]	RW	speed_stat_dio	Port speed mode set by the CPU. 0: 10 Mbit/s mode. 1: 100 Mbit/s mode.
[1]	RW	link_stat_dio	Port link status set by the CPU. 0: No link exists. 1: A link exists.
[0]	RW	duplex_stat_dio	Port duplex mode set by the CPU. 0: Half-duplex. 1: Full-duplex.

UD_MAC_STAT_CHANGE

UD_MAC_STAT_CHANGE is the port status change indicator register. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x020C	UD_MAC_STAT_CHANGE	0x0000_0000

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

Bits	Access	Name	Description
[31:3]	RO	reserved	Reserved.
[2]	WC	speed_stat_ch	Port speed mode change indicator. 0: No change occurs. 1: A change occurs. Writing 1 clears this register.
[1]	WC	link_stat_ch	Port link status change indicator. 0: No change occurs. 1: A change occurs. Writing 1 clears this register.

[0]	WC	duplex_stat_ch	Port duplex mode change indicator. 0: No change occurs. 1: A change occurs. Writing 1 clears this register.

UD_MAC_SET

UD_MAC_SET is the MAC function configuration register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0210	UD_MAC_SET	0x2027_55EE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved		add_pad_en	crcgen_dis	cntr_rdclr_en	cntr_clr_all	cntr_roll_dis	colthreshold				in_loop_en	ex_loop_en	pause_en	rx_shframe_en	rx_min_thr					len_max											
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1	0	1	0	1	0	1	1	1	1	0	1	1	1	0

Bits	Access	Name	Description
[31:30]	RO	reversed	Reserved.
[29]	RW	add_pad_en	Port auto add PAD enable during transmission. 0: Disabled. 1: Enabled.
[28]	RW	crcgen_dis	Port CRC generation disable control. 0: Transmit frame recalculate CRC. 1: Transmit frame not recalculate CRC.
[27]	RW	cntr_rdclr_en	Port statistics counter read clear enable. 0: Disabled. 1: Enabled.
[26]	RW	cntr_clr_all	Port statistics counter clear control. 0: Not clear. 1: Clear. **Note:** 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.
[25]	RW	cntr_roll_dis	Port statistics acyclic counter enable. 0: Disabled. 1: Enabled.

[24:21]	RW	colthreshold	Port collision count statistics threshold. The default value is 0x1, which indicates the count of frames with one collision.
[20]	RW	in_loop_en	Port loopback to internal enable. 0: Disabled. 1: Enabled. **Note:** Loopback to internal enable and loopback to external enable cannot be configured at the same time. When the network interface is in normal state, you need to perform soft reset on the module after loopback to internal enable is configured instead of loopback to external enable and vice versa.
[19]	RW	ex_loop_en	Port loopback to external enable. 0: Disabled. 1: Enabled. **Note:** Loopback to internal enable and loopback to external enable cannot be configured at the same time. When the network interface is in normal state, you need to perform soft reset on the module after loopback to internal enable is configured instead of loopback to external enable and vice versa.
[18]	RW	pause_en	Port pause frame transmit enable. 0: Disabled. 1: Enabled.
[17]	RW	rx_shframe_en	Port short frame receive enable. 0: Disabled. 1: Enabled. **Note:** If rx_shframe_en is set to 1, the minimum receive frame length allowed by the port is that set by rx_min_thr. If rx_shframe_en is set to 0, the minimum receive frame length allowed by the port is 64 bytes (including CRC) by default.
[16:11]	RW	rx_min_thr	Minimum receive frame length allowed by the port. The value range is from 42 bytes to 63 bytes. The default value is 42 bytes. **Note:** If rx_min_thr is set to a value smaller than 42, 42 is used instead of the value.
[10:0]	RW	len_max	Maximum receive frame length allowed by the port. The default value is 1518 bytes. The value is in a range of 1518 bytes to 1535 bytes. **Note:** 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.

5.6.3 Description of the Global Control Registers

GLB_HOSTMAC_L32

GLB_HOSTMAC_L32 is the lower 32-bit register for the local MAC address.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x1300	GLB_HOSTMAC_L32	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	local_mac	Lower 32 bits of the local MAC address.

GLB_HOSTMAC_H16

GLB_HOSTMAC_H16 is the upper 16-bit register for the local MAC address.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x1304	GLB_HOSTMAC_H16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																local_mac[47:32]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	RW	local_mac[47:32]	Upper 16 bits of the local MAC address.

GLB_SOFT_RESET

GLB_SOFT_RESET is the 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	Register Name	Total Reset Value
0x1308	GLB_SOFT_RESET	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RW	soft_reset	Internal soft reset. 0: Not reset. 1: Reset. In soft reset state, this bit must be set to 0to clear soft reset.

GLB_FWCTRL

GLB_FWCTRL is the forward control register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x1310	GLB_FWCTRL	0x0000_0020

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7	6	5	4	3 2 1 0
Name	reserved						fwall2cpu_up	reserved	fw2cpu_ena_up	reserved	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0	0	1	0	0 0 0 0

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[7]	RW	fwall2cpu_up	Indicates whether to forcibly forward all valid input frames to the CPU port. 0: no 0: yes
[6]	RO	reserved	Reserved.
[5]	RW	fw2cpu_ena_up	Function enable of forwarding the input frames to the CPU port. 0: disabled 1: enabled

[4:0]	RO	reserved	Reserved.

GLB_MACTCTRL

GLB_MACTCTRL is the 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	Register Name	Total Reset Value
0x1314	GLB_MACTCTRL	0x0000_0020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								mact_ena_up	reserved	broad2cpu_up	reserved	multi2cpu_up	reserved	uni2cpu_up	reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Access	Name	Description
[31:8]	RO	reserved	Reserved.
[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).
[6]	RO	reserved	Reserved.
[5]	RW	broad2cpu_up	Indicates whether to forward the input broadcast frames to the CPU port. 0: no 1: yes
[4]	RO	reserved	Reserved.
[3]	RW	multi2cpu_up	Indicates whether to forward the input multicast frames that are not listed in the filter table to the CPU port. 0: no 1: yes
[2]	RO	reserved	Reserved.

[1]	RW	uni2cpu_up	Indicates whether to forward the input unicast frames that are not listed in the filter table to the CPU port. 0: no 1: yes
[0]	RO	reserved	Reserved.

GLB_ENDIAN_MOD

GLB_ENDIAN_MOD is the endian control register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x1318	GLB_ENDIAN_MOD	0x0000_0003

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1]	RW	in_edian	Receive packet write SDRAM endian configuration. 0: Big-endian mode. 1: Little-endian mode. Data consists of bytes.
[0]	RW	out_edian	Transmit packet read SDRAM endian configuration. 0: Big-endian mode. 1: Little-endian mode.

GLB_IRQ_STAT

GLB_IRQ_STAT is the interrupt status register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x1330	GLB_IRQ_STAT	0x0000_0000

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

Name	reserved	int_mdio_finish	reserved	int_rxd_up	int_freeeq_up	int_stat_up	int_duplex_up	int_speed_up	int_link_up	int_tx_up	int_rx_up
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0 0 0 0 0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:13]	RO	reserved	Reserved.
[12]	RO	int_mdio_finish	Interrupt status indicates whether the MDIO interface completes the operation required by the CPU. 0: Not completed. 1: Completed and An interrupt occurs. After this interrupt is generated, software determines whether the MDIO completes the operation by querying MDIO_RWCTRL[finish].
[11:8]	RO	reserved	Reserved.
[7]	RO	int_rxd_up	Interrupt status (multi-packet interrupt) for a frame (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 receive queue. After this interrupt is generated, software determines whether there are frames to be received by querying GLB_IRQ_RAW[iraw_rxd_up].
[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 transmit queue buffer is changed from nonempty to empty, that is, the status of the transmit 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 occurs. 1: An interrupt occurs. After this interrupt is generated, software determines whether the current transmit queue is empty by querying UD_GLB_ADDRQ_STAT[eq_cnt]. If the current transmit 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 UD_MDIO_IRQENA.

[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 UD_MAC_STAT_CHANGE[duplex_stat_ch].
[3]	RO	int_speed_up	Interrupt status for ort 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 UD_MAC_STAT_CHANGE[speed_stat_ch].
[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 changes. After this interrupt is generated, software determines whether the link status changes by querying UD_MAC_STAT_CHANGE[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 occurs. After this interrupt is generated, software determines whether to release the buffer of the transmit frames by querying the current transmit queue address eq_out_index in UD_GLB_QSTAT.
[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 receive queue. After this interrupt program is started, software determines whether frames are received by querying the GLB_IRQ_RAW[iraw_rxd_up] signal.

GLB_IRQ_ENA

GLB_IRQ_ENA is the interrupt enable register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x1334	GLB_IRQ_ENA	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												ien_all	ien_up	reserved					ien_mdio_finish	reserved				ien_rxd_up	ien_freeeq_up	ien_stat_up	ien_duplex_up	ien_speed_up	ien_link_up	ien_tx_up	ien_rx_up
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:20]	RO	reserved	Reserved.
[19]	RW	ien_all	All interrupts enable. 0: Disabled (none of the interrupt can be reported). 1: Enabled (all interrupts are reported 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).
[17:13]	RO	reserved	Reserved.
[12]	RW	ien_mdio_finish	Indicator enable for the MDIO to complete the operation required by the CPU. 0: Disabled. 1: Enabled.
[11:8]	RO	reserved	Reserved.
[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.
[6]	RW	ien_freeeq_up	Interrupt signal enable for the transmit queue of the uplink port to change from nonempty to empty. 0: Disabled. 1: Enabled.
[5]	RW	ien_stat_up	Interrupt signal enable for uplink port status changes. 0: Disabled. 1: Enabled.

[4]	RW	ien_duplex_up	Interrupt enable for uplink port duplex mode changes. 0: Disabled. 1: Enabled.
[3]	RW	ien_speed_up	Interrupt enable for uplink port speed mode changes. 0: Disabled. 1: Enabled.
[2]	RW	ien_link_up	Interrupt enable for uplink port link 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.
[0]	RW	ien_rx_up	Interrupt enable for frames on the uplink port to be received by the CPU. 0: Disabled. 1: Enabled.

GLB_IRQ_RAW

GLB_IRQ_RAW is the raw interrupt register. The register does not support soft reset. Writing 1 clears this register.

Offset Address	Register Name	Total Reset Value
0x1338	GLB_IRQ_RAW	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13	12	11 10 9 8	7	6	5	4	3	2	1	0
Name	reserved					iraw_mdio_finish	reserved	iraw_rxd_up	iraw_freeeq_up	iraw_stat_up	iraw_duplex_up	iraw_speed_up	iraw_link_up	iraw_tx_up	iraw_rx_up
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0	0	0 0 0 0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:13]	RO	reserved	Reserved.
[12]	WC	iraw_mdio_finish	Raw interrupt status for the MDIO to complete the operation required by the CPU. 0: No interrupt occurs. 1: An interrupt occurs.
[11:8]	RO	reserved	Reserved.

[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 occurs. 1: An interrupt occurs.
[6]	WC	iraw_freeeq_up	Raw interrupt status for the transmit queue of the uplink port to change from nonempty to empty, indicating that the transmit queue buffer changes from nonempty to empty and the CPU can write a group of new frames to be transmitted. 0: No interrupt occurs. 1: An interrupt occurs.
[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 occurs. 1: An interrupt occurs.
[4]	WC	iraw_duplex_up	Raw interrupt status for uplink port duplex mode changes. 0: No interrupt occurs. 1: An interrupt occurs.
[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.
[2]	WC	iraw_link_up	Raw interrupt status for uplink port link status changes. 0: No interrupt occurs. 1: An interrupt occurs.
[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 occurs. 1: An interrupt occurs.
[0]	WC	iraw_rx_up	Raw interrupt status for frames on the uplink port to be received by the CPU. 0: No interrupt occurs. 1: An interrupt occurs.

GLB_MAC0_L32

GLB_MAC0_L32 is the lower 32-bit register for the filter table MAC0.

Offset Address	Register Name	Total Reset Value
0x1400	GLB_MAC0_L32	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	flt_mac0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	flt_mac0	Lower 32 bits of the filter table MAC0.

GLB_MAC0_H16

GLB_MAC0_H16 is the upper 16-bit register for the filter table MAC0.

Offset Address	Register Name	Total Reset Value
0x1404	GLB_MAC0_H16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										fw2cpu_up	reserved			mac0_up	reserved	flt_mac0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RW	fw2cpu_up	Indicates 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
[20:18]	RO	reserved	Reserved.
[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.
[16]	RO	reserved	Reserved.
[15:0]	RW	flt_mac0	Upper 16 bits of the filter table MAC0.

GLB_MAC1_L32

GLB_MAC1_L32 is the lower 32-bit register for the filter table MAC1.

Offset Address	Register Name	Total Reset Value
0x1408	GLB_MAC1_L32	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	flt_mac1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	flt_mac1	Lower 32 bits of the filter table MAC1.

GLB_MAC1_H16

GLB_MAC1_H16 is the upper 16-bit register for the filter table MAC1.

Offset Address	Register Name	Total Reset Value
0x140C	GLB_MAC1_H16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										fw2cpu_up	reserved			mac1_up	reserved	flt_mac1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RW	fw2cpu_up	Indicates 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
[20:18]	RO	reserved	Reserved.
[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.
[16]	RO	reserved	Reserved.
[15:0]	RW	flt_mac1	Upper 16 bits of the filter table MAC1.

GLB_MAC2_L32

GLB_MAC2_L32 is the lower 32-bit register for the filter table MAC2.

Offset Address	Register Name	Total Reset Value
0x1410	GLB_MAC2_L32	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	flt_mac2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	flt_mac2	Lower 32 bits of the filter table MAC2.

GLB_MAC2_H16

GLB_MAC2_H16 is the upper 16-bit register for the filter table MAC2.

Offset Address	Register Name	Total Reset Value
0x1414	GLB_MAC2_H16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										fw2cpu_up		reserved				mac2_up		reserved		flt_mac2											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RW	fw2cpu_up	Indicates 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
[20:18]	RO	reserved	Reserved.
[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.
[16]	RO	reserved	Reserved.
[15:0]	RW	flt_mac2	Upper 16 bits of the filter table MAC2.

GLB_MAC3_L32

GLB_MAC3_L32 is the lower 32-bit register for the filter table MAC3.

Offset Address	Register Name	Total Reset Value
0x1418	GLB_MAC3_L32	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	flt_mac3																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	flt_mac3	Lower 32 bits of the filter table MAC3.

GLB_MAC3_H16

GLB_MAC3_H16 is the upper 16-bit register for the filter table MAC3.

Offset Address	Register Name	Total Reset Value
0x141C	GLB_MAC3_H16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										fw2cpu_up	reserved			mac3_up	reserved	flt_mac3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RW	fw2cpu_up	Indicates 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
[20:18]	RO	reserved	Reserved.
[17]	RW	mac3_up	Indicates 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.
[16]	RO	reserved	Reserved.
[15:0]	RW	flt_mac3	Upper 16 bits of the filter table MAC3.

GLB_MAC4_L32

GLB_MAC4_L32 is the lower 32-bit register for the filter table MAC4.

Offset Address	Register Name	Total Reset Value
0x1420	GLB_MAC4_L32	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	flt_mac4																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	flt_mac4	Lower 32 bits of the filter table MAC4.

GLB_MAC4_H16

GLB_MAC4_H16 is the upper 16-bit register for the filter table MAC4.

Offset Address	Register Name	Total Reset Value
0x1424	GLB_MAC4_H16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										fw2cpu_up	reserved			mac4_up	reserved	flt_mac4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RW	fw2cpu_up	Indicates 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
[20:18]	RO	reserved	Reserved.
[17]	RW	mac4_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.
[16]	RO	reserved	Reserved.
[15:0]	RW	flt_mac4	Upper 16 bits of the filter table MAC4.

GLB_MAC5_L32

GLB_MAC5_L32 is the lower 32-bit register for the filter table MAC5.

Offset Address	Register Name	Total Reset Value
0x1428	GLB_MAC5_L32	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	flt_mac5																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	flt_mac5	Lower 32 bits of the filter table MAC5.

GLB_MAC5_H16

GLB_MAC5_H16 is the upper 16-bit register for the filter table MAC5.

Offset Address	Register Name	Total Reset Value
0x142C	GLB_MAC5_H16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										fw2cpu_up	reserved			mac5_up	reserved	flt_mac5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RW	fw2cpu_up	Indicates 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
[20:18]	RO	reserved	Reserved.
[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.
[16]	RO	reserved	Reserved.
[15:0]	RW	flt_mac5	Upper 16 bits of the filter table MAC5.

GLB_MAC6_L32

GLB_MAC6_L32 is the lower 32-bit register for the filter table MAC6.

Offset Address	Register Name	Total Reset Value
0x1430	GLB_MAC6_L32	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	flt_mac6							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:0]	RW	flt_mac6	Lower 32 bits of the filter table MAC6.

GLB_MAC6_H16

GLB_MAC6_H16 is the upper 16-bit register for the filter table MAC6.

Offset Address	Register Name	Total Reset Value
0x1434	GLB_MAC6_H16	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved				flt_mac6			
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RW	fw2cpu_up	Indicates 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
[20:18]	RO	reserved	Reserved.
[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.
[16]	RO	reserved	Reserved.
[15:0]	RW	flt_mac6	Upper 16 bits of the filter table MAC6.

GLB_MAC7_L32

GLB_MAC7_L32 is the lower 32-bit register for the filter table MAC7.

Offset Address	Register Name	Total Reset Value
0x1438	GLB_MAC7_L32	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	flt_mac7																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	flt_mac7	Lower 32 bits of the filter table MAC7.

GLB_MAC7_H16

GLB_MAC7_H16 is the upper 16-bit register for the filter table MAC7.

Offset Address	Register Name	Total Reset Value
0x143C	GLB_MAC7_H16	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										fw2cpu_up	reserved			mac7_up	reserved	flt_mac7															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RW	fw2cpu_up	Indicates 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
[20:18]	RO	reserved	Reserved.
[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.
[16]	RO	reserved	Reserved.
[15:0]	RW	flt_mac7	Upper 16 bits of the filter table MAC7.

UD_GLB_IRQN_SET

UD_GLB_IRQN_SET is the multi-packet interrupt configuration register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0340	UD_GLB_IRQN_SET	0x0800_003A

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

Bits	Access	Name	Description
[31:29]	RO	reserved	Reserved.
[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. **Note:** The minimum value of int_frm_cnt can be set to 1. In this case, multi-packet interrupt is equivalent to single-packet interrupt.
[23:16]	RO	reserved	Reserved.
[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. **Note:** age_timer is counted in the unit of the main clock cycle divided by 256.

UD_GLB_QLEN_SET

UD_GLB_QLEN_SET is the queue length configuration register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0344	UD_GLB_QLEN_SET	0x0000_2020

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

Bits	Access	Name	Description
[31:14]	RO	reserved	Reserved.

[13:8]	RW	iq_len	Receive (packet receive) queue length configuration. **Note:** 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 firstly assigned and the value of eq_len is calculated by the formula: 64 – iq_len.
[7:6]	RO	reserved	Reserved.
[5:0]	RW	eq_len	Transmit (packet transmit) queue length configuration. **Note:** eq_len cannot be set to 0. Otherwise, it is forcibly set to 1.

UD_GLB_FC_LEVEL

UD_GLB_FC_LEVEL is the traffic control register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0348	UD_GLB_FC_LEVEL	0x3018_0508

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

Bits	Access	Name	Description
[31:15]	RO	reserved	Reserved.
[14]	RW	qlimit_ena	Traffic control enable for receive queue. 0: Disabled (do not transmit the traffic control message according to the status of receive queue). 1: Enabled (transmit the traffic control message according to the status of receive queue).
[13:8]	RW	qlimit_up	Upper limit of traffic control for receive queue. When the free space of the receive queue is less than the upper limit, if traffic control for receive queue is enabled, the traffic control message is transmitted to the peer end. **Note:** If the upper limit qlimit_up is set to 0, the receive queue fails to enter the traffic control status. The upper limit qlimit_up must be greater than the lower limit qlimit_down.
[7:6]	RO	reserved	Reserved.

[5:0]	RW	qlimit_down	Lower limit of traffic control for receive queue. When the free space of the receive queue is equal to or greater than the upper limit, if the receive queue is in traffic control state, the current traffic control is stopped.

UD_GLB_CAUSE

UD_GLB_CAUSE is the cause register for the CPU to which the packet is transmitted.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x034C	UD_GLB_CAUSE	0x0000_0000

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

Bits	Access	Name	Description
[31:3]	RO	reserved	Reserved.
[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.

UD_GLB_RXFRM_SADDR

UD_GLB_RXFRM_SADDR is the receive frame start address register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0350	UD_GLB_RXFRM_SADDR	0x0000_0000

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

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	rxfrm_saddr	Start address of the receive frame.

UD_GLB_IQFRM_DES

UD_GLB_IQFRM_DES is the receive frame descriptor register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0354	UD_GLB_IQFRM_DES	0x0000_0000

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

Bits	Access	Name	Description
[31:18]	RO	reserved	Reserved.
[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.
[11:0]	RO	fd_in_len	Length of the frame to be received in the receive queue.

UD_GLB_IQ_ADDR

UD_GLB_IQ_ADDR is the receive 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 receive frame to the 0xF000_8002 address, writes the second byte (valid data) of the receive frame to the 0xF000_8003 address. Subsequent data is written to the buffer in sequence. If the configured header address of the receive frame is other non-word-aligned address, the logic writes data in a similar way.

Offset Address	Register Name	Total Reset Value
0x0358	UD_GLB_IQ_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	startaddr_iq	Header address (configured by the CPU) of the storage space corresponding to the receive frame. The receive frame requests the bus according to this address.

UD_GLB_BFC_STAT

UD_GLB_BFC_STAT is the counter for traffic control status of forward buffer and aging time of multi-packet interrupt.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x035C	UD_GLB_BFC_STAT	0x0000_0000

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

Bits	Access	Name	Description
[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 unit of time, it indicates that UD_GLB_IRQN_SET[int_frm_cnt] is set improperly. Multi-packet interrupt is triggered by the aging time. Therefore, the configured value must be reduced.
[15:0]	RO	flowctrl_cnt	Register for the count of the forward buffer of the uplink or downlink port entering the traffic control status. **Note:** If the value of flowctrl_cnt is too large in a unit of time, it indicates that UD_GLB_FC_LEVEL[blimit_up] or UD_GLB_FC_LEVEL[blimit_down] is set to a too small value, or the external network condition is worsened. In this case, the configured value may be reduced.

UD_GLB_EQ_ADDR

UD_GLB_EQ_ADDR is the transmit queue header address register.

The register does not support soft reset.

NOTE

If the header address of the transmit 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 transmit 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 transmit frame and considers the data read from the 0xF000_8103 address as the second byte (valid data) of the transmit frame. All subsequent data is valid (until the data of the specified frame length is read). If the configured header address of the transmit frame is other non-word-aligned address, the logic reads data in a similar way.

Offset Address	Register Name	Total Reset Value
0x0360	UD_GLB_EQ_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	add_fd_addr_out	Header address of the transmit frame added by the CPU to the transmit queue.

UD_GLB_EQFRM_LEN

UD_GLB_EQFRM_LEN is the transmit queue frame length configuration register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0364	UD_GLB_EQFRM_LEN	0x0000_0000

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

Bits	Access	Name	Description
[31:11]	RO	reserved	Reserved.
[10:0]	RW	add_fd_len_out	Length of the transmit frame added by the CPU to the transmit queue. Configure this register to trigger hardware so that software can write the header address and length of the transmit frame to the transmit 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.

UD_GLB_QSTAT

UD_GLB_QSTAT is the queue status register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0368	UD_GLB_QSTAT	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				iq_in_index				reserved				cpuw_index				reserved				eq_in_index				reserved				eq_out_index			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:30]	RO	reserved	Reserved.
[29:24]	RO	iq_in_index	Receive index of the receive (packet receive) queue.
[23:22]	RO	reserved	Reserved.
[21:16]	RO	cpuw_index	Receive index of frame header address of the receive (packet receive) queue.
[15:14]	RO	reserved	Reserved.
[13:8]	RO	eq_in_index	Receive index of frame descriptor of the transmit (packet transmit) queue.
[7:6]	RO	reserved	Reserved.
[5:0]	RO	eq_out_index	Transmit index of frame descriptor of the transmit (packet transmit) queue.

UD_GLB_ADDRQ_STAT

UD_GLB_ADDRQ_STAT is the address queue status register.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x036C	UD_GLB_ADDRQ_STAT	0x0300_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						cpuaddr_in_rdy	eq_in_rdy	reserved				cpu_cnt				reserved				iq_cnt				reserved				eq_cnt			
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25]	RO	cpuaddr_in_rdy	Indicates whether the CPU can configure the frame header address of the receive queue. 0: The CPU cannot configure the frame header address of the receive queue. 1: The CPU can configure the frame header address of the receive queue. **Note:** 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.
[24]	RO	eq_in_rdy	Indicates whether the CPU can configure the frame descriptor (header address and length) of the transmit queue. 0: The CPU cannot configure the frame descriptor (header address and length) of the transmit queue. 1: The CPU can configure the frame descriptor (header address and length) of the transmit queue.
[23:22]	RO	reserved	Reserved.
[21:16]	RO	cpu_cnt	Header address count for available frames assigned by the CPU to the receive queue.
[15:14]	RO	reserved	Reserved.
[13:8]	RO	iq_cnt	Used length of the receive queue (0 to iq_len).
[7:6]	RO	reserved	Reserved.
[5:0]	RO	eq_cnt	Used length of the transmit queue (0 to eq_len).

UD_GLB_FC_TIMECTRL

UD_GLB_FC_TIMECTRL is the traffic control time configuration register.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0370	UD_GLB_FC_TIMECTRL	0x07FF_86A0

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

Bits	Access	Name	Description
[31:27]	RO	reserved	Reserved.

[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.
[16:0]	RW	flux_timer_inter	Traffic limit time slot counter, which is used to count the main clock. The default count is 100,000. For a 100-MHz main clock, the time slot is 1 ms.

UD_GLB_FC_RXLIMIT

UD_GLB_FC_RXLIMIT is the traffic control limit configuration register.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0374	UD_GLB_FC_RXLIMIT	0x0000_0000

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

Bits	Access	Name	Description
[31:20]	RO	reserved	Reserved.
[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 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 the packet drop control register for traffic limit.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0378	UD_GLB_FC_DROPCTRL	0x0000_0000

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

Bits	Access	Name	Description
[31:3]	RO	reserved	Reserved.

[2]	RW	flux_uni	Indicates whether unicast packets are discarded when the upper threshold of traffic limit is exceeded. 0: Do not discard unicast packets. 1: Discard unicast packets.
[1]	RW	flux_multi	Indicates whether multicast packets are discarded when the upper threshold of traffic limit is exceeded. 0: Do not discard multicast packets. 1: Discard multicast packets.
[0]	RW	flux_broad	Indicates whether broadcast packets are discarded when the upper threshold of traffic limit is exceeded. 0: Do not discard broadcast packets. 1: Discard broadcast packets.

5.6.4 Description of the Statistics Counter Control Registers

UD_STS_PORTCNT

UD_STS_PORTCNT is the port status counter.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0584	UD_STS_PORTCNT	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	rxsof_cnt	rxeof_cnt	rxcrcok_cnt	rxcrcbad_cnt	txsof_cnt	txeof_cnt	txcrcok_cnt	txcrcbad_cnt
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:28]	RO	rxsof_cnt	Count of the frame headers received by the port.
[27:24]	RO	rxeof_cnt	Count of the frame trailers received by the port.
[23:20]	RO	rxcrcok_cnt	Count of the frames without CRC errors received by the port.
[19:16]	RO	rxcrcbad_cnt	Count of the frames with CRC errors received by the port.
[15:12]	RO	txsof_cnt	Count of the frame headers transmitted by the port.
[11:8]	RO	txeof_cnt	Count of the frame trailers transmitted by the port.
[7:4]	RO	txcrcok_cnt	Count of the frames without CRC errors transmitted by the port.
[3:0]	RO	txcrcbad_cnt	Count of the frames with CRC errors transmitted by the port.

UD_PORT2CPU_PKTS

UD_PORT2CPU_PKTS is the 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	Register Name	Total Reset Value
0x05A0	UD_PORT2CPU_PKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[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.

UD_CPU2IQ_ADDRCNT

UD_CPU2IQ_ADDRCNT is the register for the count of configuring packet receiving address queue by the CPU.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x05A4	UD_CPU2IQ_ADDRCNT	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	WC	addr_cpu	Count of configuring packet receiving address queue by the CPU successfully. Writing 0 clears this register. Writing 1 has no effect.

UD_RX_IRQCNT

UD_RX_IRQCNT is the register for the count of reporting single-packet interrupt by the uplink or downlink port.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x05A8	UD_RX_IRQCNT	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	WC	pkts_port	Count of the frame receive interrupts reported by the uplink or downlink port. Writing 0 clears this register. Writing 1 has no effect.

UD_CPU2EQ_PKTS

UD_CPU2EQ_PKTS is the 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	Register Name	Total Reset Value
0x05AC	UD_CPU2EQ_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																pkts_cpu2tx															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[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.

5.6.5 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 the RXDV rising edge count register.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0600	UD_rx_DVCNT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	rxdvrise	Count of all RXDV rising edges.

UD_RX_OCTS

UD_RX_OCTS is the register for the total number of bytes received.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0604	UD_RX_OCTS	0x0000_0000

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

Bits	Access	Name	Description
[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.

UD_RX_RIGHTOCTS

UD_RX_RIGHTOCTS is the register for the total number of bytes of received correct packets.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0608	UD_RX_RIGHTOCTS	0x0000_0000

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

Bits	Access	Name	Description

[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.

UD_HOSTMAC_PKTS

UD_HOSTMAC_PKTS is the register for the number of packets matching the local MAC address.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x060C	UD_HOSTMAC_PKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	local_mac_match	Count of correct receive 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 transmit frames.

UD_RX_RIGHTPKTS

UD_RX_RIGHTPKTS is the register for the total number of packets received by the port.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0610	UD_RX_RIGHTPKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	pkts	Count of all frames.

UD_RX_BROADPKTS

UD_RX_BROADPKTS is the register for the number of correct broadcast packets.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0614	UD_RX_broadpkts	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	broadcastpkts	Count of broadcast frames with valid length and without CRC errors, excluding pause frames and error transmit frames.

UD_RX_MULTPKTS

UD_RX_MULTPKTS is the register for the number of correct multicast packets.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0618	UD_RX_multpkts	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	multicastpkts	Count of multicast frames with valid length and without CRC errors, excluding pause frames and error transmit frames.

UD_RX_UNIPKTS

UD_RX_UNIPKTS is the register for the number of correct unicast packets.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x061C	UD_RX_UNIpkts	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	ifinucastpkts	Count of unicast frames with valid length and without CRC errors, excluding pause frames and error transmit frames.

UD_RX_ERRPKTS

UD_RX_ERRPKTS is the register for the total number of incorrect packets.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0620	UD_RX_ERRPKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	ifinerrors	Count of all error frames, including frames with CRC errors, short frames, long frames, and error transmit frames.

UD_RX_CRCERR_PKTS

UD_RX_CRCERR_PKTS is the register for the count of CRC errors.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0624	UD_RX_crcerr_PKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	crcerr	Count of receive frames with valid length (non short and long frames) but with CRC or alignment errors.

UD_RX_LENERR_PKTS

UD_RX_LENERR_PKTS is the register for the number of packets with invalid length.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0628	UD_RX_LENERR_pkts	0x0000_0000

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

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[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).

UD_RX_OCRCERR_PKTS

UD_RX_OCRCERR_PKTS is the register for the number of packets with odd nibbles and CRC errors.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x062C	UD_RX_OCRCERR_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dot3alignmenterr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	dot3alignmenterr	Received frames with odd nibbles and CRC errors.

UD_RX_PAUSE_PKTS

UD_RX_PAUSE_PKTS is the register for the number of received pause packets.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0630	UD_RX_PAUSE_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dot3pause																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	dot3pause	Count of received pause frames.

UD_RF_OVERCNT

UD_RF_OVERCNT is the register for the count of RXFIFO overflow events.

The register does not support soft reset.

offset Address	Register Name	Total Reset Value
0x0634	UD_RF_OVERCNT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	dropevents	Accumulative count of RXFIFO overflow events during the reception of frames.

UD_FLUX_TOL_IPKTS

UD_FLUX_TOL_IPKTS is the register for the total number of received packets allowed by the traffic limit.

The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0638	UD_flux_TOL_IPKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	flux_frame_cnt	Total count of correct receive frames allowed by the traffic limit, excluding short frames, long frames, frames with CRC errors, pause frames, and error transmit frames.

UD_FLUX_TOL_DPKTS

UD_FLUX_TOL_DPKTS is the register for the total number of packets discarded due to traffic limit. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x063C	UD_flux_TOL_DPKTS	0x0000_0000

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

Bits	Access	Name	Description

[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 transmit frames.

UD_VN2OTH_PKTS

UD_VN2OTH_PKTS is the register for the number of packets not forwarded to another port due to VLAN limit. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0640	UD_VN2OTH_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vlan_not2oth_pkts																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	vlan_not2oth_pkts	Number of packets not forwarded to another port due to VLAN limit.

UD_VN2CPU_PKTS

UD_VN2CPU_PKTS is the register for the number of packets not forwarded to the CPU port due to VLAN limit. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0644	UD_VN2CPU_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vlan_not2cpu_pkts																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	vlan_not2cpu_pkts	Number of packets not forwarded to the CPU port due to VLAN limit.

UD_MN2OTH_PKTS

UD_MN2OTH_PKTS is the register for the number of packets not forwarded to another port due to MAC limit. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0648	UD_MN2OTH_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	mac_not2oth_pkts																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	mac_not2oth_pkts	Number of packets not forwarded to another port due to MAC limit.

UD_MN2CPU_PKTS

UD_MN2CPU_PKTS is the 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	Register Name	Total Reset Value
0x064C	UD_MN2CPU_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	mac_not2cpu_pkts																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	mac_not2cpu_pkts	Number of packets not forwarded to the CPU port due to MAC limit.

UD_TX_PKTS

UD_TX_PKTS is the register for the total number of packets transmitted successfully. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0780	UD_TX_PKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	pkts_tx	Count of all configured transmit frames, excluding the frames discarded due to timeout and the transmit frames whose length of UD_GLB_EQFRM_LEN is not within valid range.

UD_TX_BROADPKTS

UD_TX_BROADPKTS is the register for the number of broadcast packets transmitted successfully. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0784	UD_TX_BROADPKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	broadcastpkts_tx	Count of broadcast frames transmitted successfully (excluding retransmission).

UD_TX_MULTPKTS

UD_TX_MULTPKTS is the register for the number of multicast packets transmitted successfully. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0788	UD_TX_MULTPKTS	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	multicastpkts_tx	Count of multicast frames transmitted successfully (excluding retransmission).

UD_TX_UNIPKTS

UD_TX_UNIPKTS is the register for the number of unicast packets transmitted successfully. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x078C	UD_TX_UNIPKTS	0x0000_0000

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

Bits	Access	Name	Description

[31:0]	RO	ifoutucastpkts_tx	Count of unicast frames transmitted successfully (excluding retransmission).

UD_TX_OCTS

UD_TX_OCTS is the register for the total number of transmitted bytes. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0790	UD_TX_OCTS	0x0000_0000

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

Bits	Access	Name	Description
[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.

UD_TX_PAUSE_PKTS

UD_TX_PAUSE_PKTS is the register for the number of transmitted pause frames. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0794	UD_TX_PAUSE_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dot3outpause																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	dot3outpause	Count of transmitted pause frames.

UD_TX_RETRYCNT

UD_TX_RETRYCNT is the register for the total count of retransmission. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x0798	UD_TX_RETRYCNT	0x0000_0000

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

Name	retry_times_tx																											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	retry_times_tx	Total count of retransmissions of transmit frames.

UD_TX_COLCNT

UD_TX_COLCNT is the register for the total count of collisions. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x079C	UD_TX_COLCNT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	collisions	Count of collisions.

UD_TX_LC_PKTS

UD_TX_LC_PKTS is the register for the number of packets with late collision. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x07A0	UD_TX_LC_PKTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dot3latecol																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	dot3latecol	Count of packets with late collision.

UD_TX_COLOK_PKTS

UD_TX_COLOK_PKTS is the register for the number of packets transmitted successfully with collisions. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x07A4	UD_TX_COLOK_PKTS	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	dot3col_ok							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:0]	RO	dot3col_ok	Count of packets transmitted successfully with collisions.

UD_TX_RETRY15_PKTS

UD_TX_RETRY15_PKTS is the register for the number of packets discarded due to more than 15 times of retransmission. The register does not support soft reset.

Offset Address	Register Name	Total Reset Value
0x07A8	UD_TX_RETRY15_PKTS	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	dot3excessivecol							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:0]	RO	dot3excessivecol	Count of the packets discarded due to more than 15 times of retransmission.

UD_TX_RETRYN_PKTS

UD_TX_RETRYN_PKTS is the 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	Register Name	Total Reset Value
0x07AC	UD_TX_RETRYN_PKTS	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	dot3colcnt							
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:0]	RO	dot3colcnt	Count of packets with the count of collisions being equal to the threshold. This register is set by UD_MAC_SET[colthreshold].

Contents

Figures

Tables

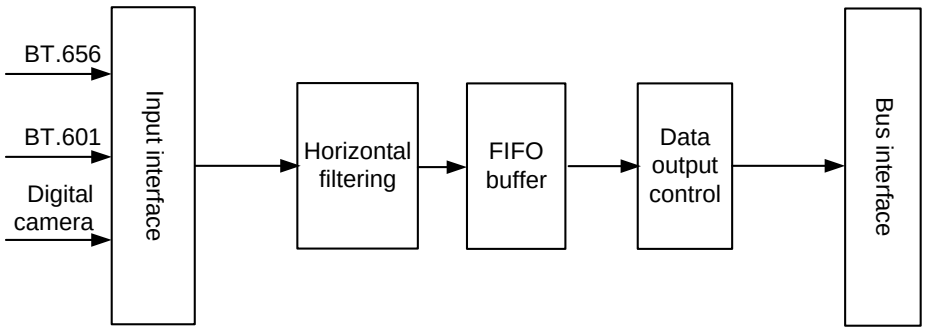
6 Video Interface

6.1 VIU

6.1.1 Overview

The video input unit (VIU) of the Hi3515 receives video data through the BT.656/601 interface and the digital camera (DC) interface, and then stores the data in a specified memory. During this process, the VIU supports the 1/2 horizontal down scaling and chrominance resampling for the video graphic data. Figure 6-1 shows the functional block diagram of the VIU.

Figure 6-1 Functional block diagram of the VIU



6.1.2 Features

The features of the VIU are as follows:

- Supports four external video ports.
- Processes the videos of eight channels inside.
- Each port supports standard BT.656 video inputs.
- Port 0 and port 2 support 2-channel or 4-channel time-division-multiplexed (TDM) video inputs.
- Port 1 and port 3 support 2-channel TDM video inputs (note that port 0 and port 2 must be in non-4-channel TDM mode in this case).
- Each port supports interlaced and progressive input modes.

- Port 0 and port 2 support BT.601 video inputs.
- Port 0 and port 2 support the connection to the DC interface (up to 3 megapixels).
- Ports 0–3 support the BT1120 high-definition (HD) interface timing.
- Ports 0–3 support the 16-bit synchronous interface in HD mode.
- Receives data by fields or by frames.
- Supports input data in four sequences: CbYCrY, CrYCbY, YCbYCr, and YCrYCb4.
- Supports 1/2 horizontal down scaling.
- Supports the conversion from the co-sited format to interspersed format during horizontal chrominance resampling.
- Supports configurable level-2 bus priorities for each channel.
- Obtains data in a specified window.
- Supports image block masking.
- Supports the following output storage modes:
  - package YCbCr 4:2:2 mode
  - SPYCbCr 4:2:0 and SPYCbCr 4:2:2 mode
- Supports the raw data storage mode during data output, for the HD Y/C separation input mode only.
- Supports statistics on image luminance and luminance stretch.
- Provides an advance high-performance bus (AHB) master interface through which the data is written to the double data-rate (DDR) directly.
- Provides an AHB slave interface through which the configuration and status information about VIU registers is read.

6.1.3 Signal Description

Table 6-1 shows the external input or output signals of the VI interface.

Table 6-1 Signals of the VI interface

Signal Name	Direction	Description	Related Pins
VI_DATAIN_P0[7:0]	I	Input data of video port 0	VI0DAT7–VI0DAT0
VI_DATAIN_P1[7:0]	I	Input data of video port 1	VI1DAT7–VI1DAT0
VI_DATAIN_P2[7:0]	I	Input data of video port 2	VI2DAT7–VI2DAT0
VI_DATAIN_P3[7:0]	I	Input data of video port 3	VI3DAT7–VI3DAT0
CLK_VI_P0	I	Input clock of video port 0	VI0CK
CLK_VI_P1	I	Input clock of video port 1	VI1CK
CLK_VI_P2	I	Input clock of video port 2	VI2CK
CLK_VI_P3	I	Input clock of video port 3	VI3CK

Signal Name	Direction	Description	Related Pins
VI_P0_HSYNC_VD	I	Horizontal sync pulse or data valid signal of port 0. The register VIn_PORT_CFG[port_hsync] controls whether this signal is a sync pulse signal or a data valid signal. port_hsync = 0: data valid level. VIn_PORT_CFG[port_hsync_neg] controls whether the level is active high or active low. port_hsync = 1: sync pulse. VIn_PORT_CFG[port_hsync_neg] controls whether the pulse is positive or negative.	VI0HS
VI_P2_HSYNC_VD	I	Horizontal sync pulse or data valid signal of port 2. The register VIn_PORT_CFG[port_hsync] controls whether this signal is sync pulse signal or a data valid signal. port_hsync = 0: data valid level. VIn_PORT_CFG[port_hsync_neg] controls whether the level is active high or active low. port_hsync = 1: sync pulse. VIn_PORT_CFG[port_hsync_neg] controls whether the pulse is positive or negative.	VI2HS
VI_P0_VSYNC_FIELD	I	Vertical sync pulse or field indicator signal of port 0. The register VIn_PORT_CFG[port_vsync] controls whether this signal is a sync pulse signal or a field indicator signal. port_vsync = 0: field indicator signal. VIn_PORT_CFG[port_vsync_neg] controls whether the level is active high or active high. Both high level and low level can indicate odd field or even field. port_hsync = 1: sync pulse. VIn_PORT_CFG[port_vsync_neg] controls whether the pulse is positive or negative.	VI0VS

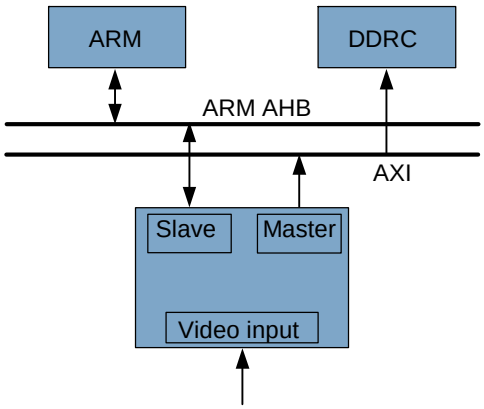
Signal Name	Direction	Description	Related Pins
VI_P2_VSYNC_FIELD	I	Vertical sync pulse or field indicator signal of port 2. The register VIn_PORT_CFG[port_vsync] controls whether this signal is a sync pulse signal or a field indicator signal. port_vsync = 0: field indicator signal. VIn_PORT_CFG[port_vsync_neg] controls whether the level is active high or active high. Both high level and low level can indicate odd field or even field. port_hsync = 1: sync pulse. VIn_PORT_CFG[port_vsync_neg] controls whether the pulse is positive or negative.	VI2VS

6.1.4 Function Description

6.1.4.1 Typical Application

Figure 6-2 shows the typical application of the VIU.

Figure 6-2 Typical application of the VIU



The VIU is a unit that collects video inputs in multiple timings and then stores video data in the DDR. By using different function modes configured by the system, the VIU can be connected to different external VI interfaces, thus supporting multiple external input devices.

6.1.4.2 Function Principle

ITU-R BT. 656 YCbCr4:2:2

1. Horizontal timings

Based on the ITU-R BT.656 Recommendation, sync signals are integrated in data streams. The special bytes start of active video (SAV) and end of active video (EAV) indicate the start and end of valid line data respectively. In a video stream, the timing reference code consisting of FF 00 00 (FF and 00 are the reserved values of encoded image data instead of image data) indicates whether its adjacent byte is SAV or EAV. Table 6-2 shows the format of the line data stream in the ITU-R BT.656 Recommendation.

Table 6-2 Format of the ITU-R BT.656 YCbCr 4:2:2 line data

Timing Reference Code				Line Blanking Area					Timing Reference Code				YCbCr 4:2:2 with 720 Valid Pixels						
FF	00	00	EAV	80	10	…	80	10	FF	00	00	SAV	Cb0	Y0	Cr0	Y1	…	Cr718	Y719

The difference between the SAV and EAV depends on the special bit H. Both the SAV and EAV include vertical blanking bit V and field indicator bit F. For details about the SAV and EAV, see Table 6-3.

Table 6-3 Formats of SAV and EAV

Bit7	Bit6(F)	Bit5(V)	Bit4(H)	Bit[3:0](P3–P0)
Fixed value 1	Field indicator bits 1st field: F = 0 2nd field: F = 1	Vertical blanking bits VBI: V = 1 Active video: V = 0	SAV: H = 0 EAV: H = 1	Check bits

The ITU-R BT.656 Recommendation defines valid SAV and EAV by using eight valid reserved bits. Four check bits are used to correct 1-bit error and detect 2-bit errors. Table 6-4 shows the valid values of SAV and EAV.

Table 6-4 Valid values of SAV and EAV

Code	Binary Value	Field Number	Vertical Blanking Interval
SAV	10000000	1	–
EAV	10011101	1	–
SAV	10101011	1	Yes
EAV	10110110	1	Yes
SAV	11000111	2	–

Code	Binary Value	Field Number	Vertical Blanking Interval
EAV	11011010	2	–
SAV	11101100	2	Yes
EAV	11110001	2	Yes

The four valid reserved bits P0, P1, P2, and P3 are used to correct errors. They are determined by the bits F, V, and H. Table 6-5 shows the ITU-R BT.656 error-correcting codes.

Table 6-5 ITU-R BT.656 error-correcting codes

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Where,

P0 = F^V^H

P1 = F^V

P2 = F^H

P3 = V^H

2. Vertical timings

The positions of the vertical timings are determined by bit F and bit V of the timing reference codes SAV and EAV. Figure 6-3 shows the vertical timing of the 525-line video system and Figure 6-4 shows the vertical timing of the 625-line video system.

Figure 6-3 Vertical timing of the 525-line 60 fields/s video system

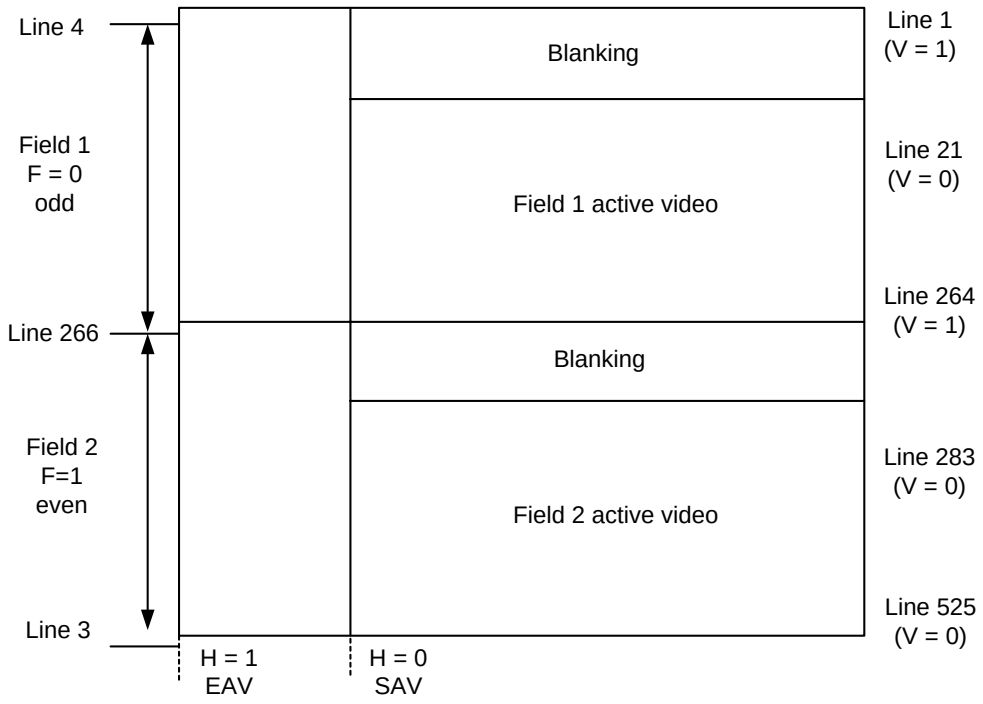
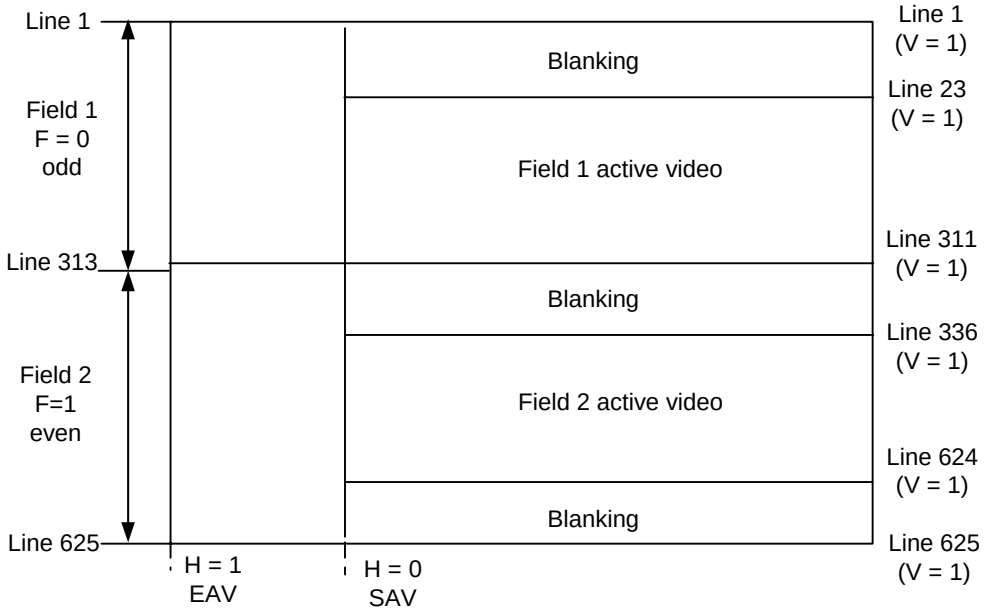


Figure 6-4 Vertical timing of the 625-line 50 fields/s video system



The VIU identifies vertical timings based on SAV and EAV regardless of the lines.

Multiple-Path BT.656 TDM Timings

Besides the standard 1-path BT.656 timing, the VIU also supports the 2-path or 4-path BT.656 timing in which data is transferred through one port. See Figure 6-5 and Figure 6-6. The data of each path is independent, but the data is time-division-multiplexed and then transmitted to the VIU through a port.

Figure 6-5 2-Path BT.656 TDM timing

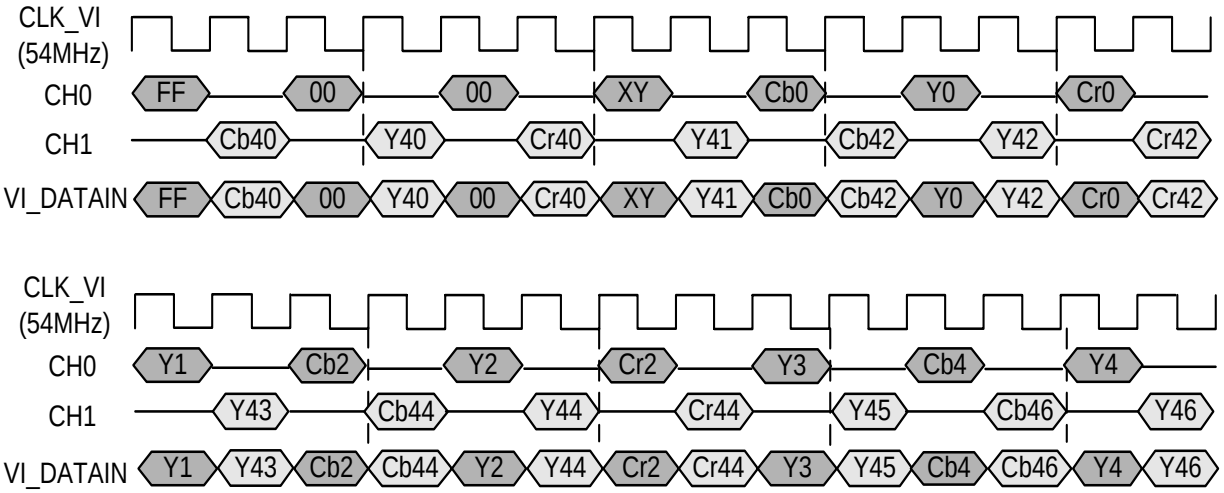
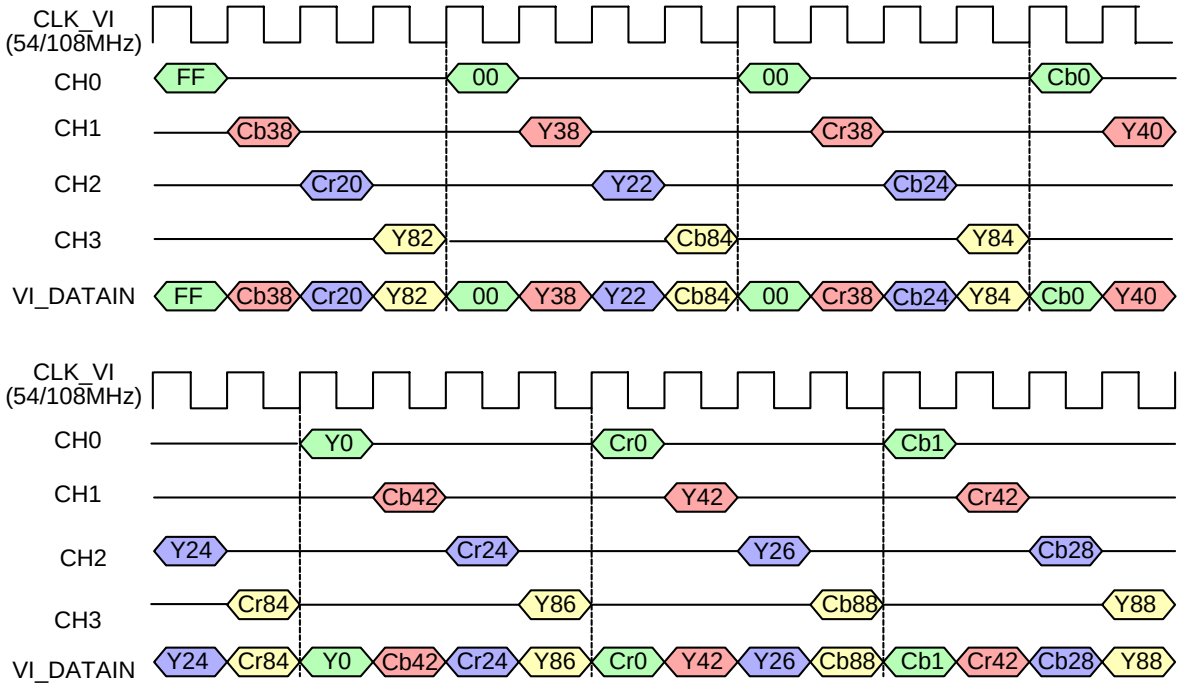


Figure 6-6 4-Path BT.656 TDM timing



When the input data is in the TDM format, the two low error-correcting bits of SAV and EAV indicate that the bit VI_CH_CFG[correct_en] must be set to 0 during input of the IDs of data channels. Table 6-6 shows the formats of SAV and EAV with channel IDs.

Table 6-6 Formats of SAV and EAV with channel IDs

Bit[7]	Bit[6]	Bit[5]	Bit[4]	Bit[3]	Bit[2]	Bit[1:0]
1	F	V	H	x	x	Channel ID.

BT 1120 (HD) Interface Timings

The VIU supports the HD interface timings with Y/C separation inputs. In this case, two ports are required. One is used to transfer luminance and the other is used to transfer chrominance, as shown in Figure 6-7 and Figure 6-8.

Figure 6-7 Horizontal input timing of the HD interface

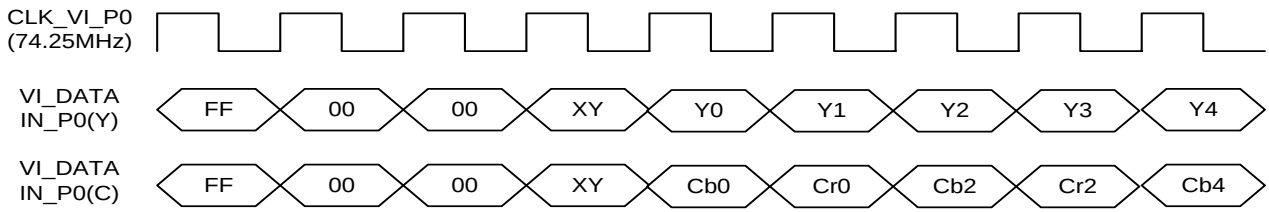
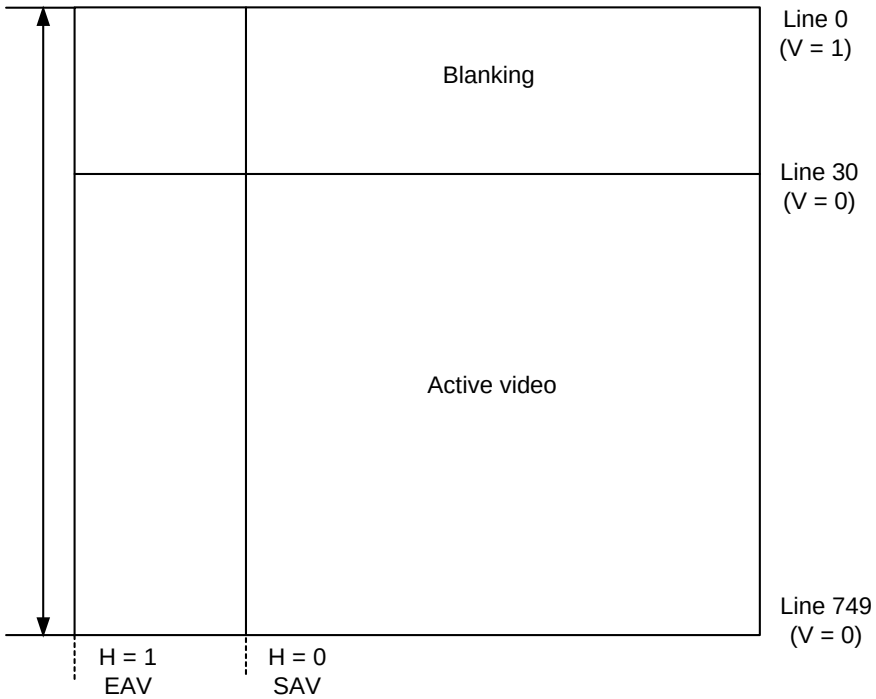


Figure 6-8 Vertical input timing of the HD interface

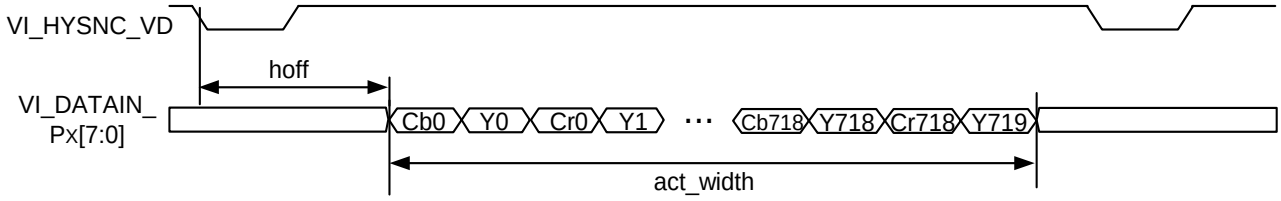


ITU-R BT.601 YCbCr4:2:2

1. Horizontal timings

The horizontal pulse indicates the start of a new line, as shown in Figure 6-9. After hoff clock cycles, an input signal passes the line front blanking area and goes to the line valid data area. The value of hoff is configurable, that is, it is 122 x 2 in phase alternating line (PAL) 525-line system and 132 x 2 in National Television Systems Committee (NTSC) 625-line system. After act_width clock cycles, the input signal passes the line valid data area and goes to the line back blanking area. The value of act_width is also configurable, that is, its typical value is 720 or 704. In addition, the horizontal sync polarity is configurable.

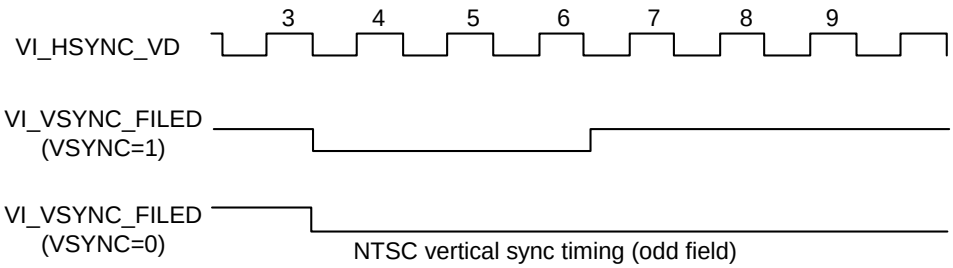
Figure 6-9 ITU-R BT.601 horizontal timing

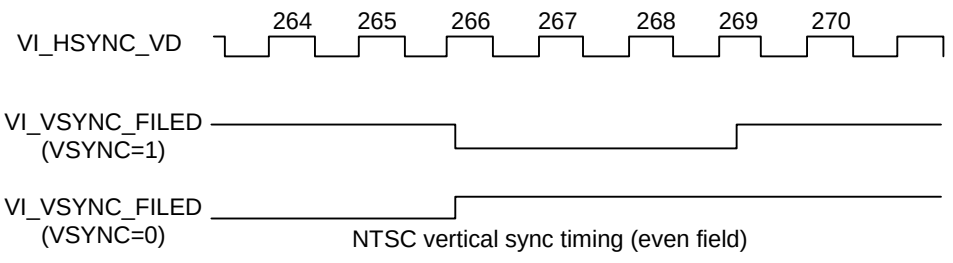


2. Vertical timings

Based on the ITU-R BT.601 recommendation, the signals VSYNC and FIELD are vertical sync signals. The VSYNC pulse or FIELD transition indicates the start of the odd field or even field. The VIU supports two types of vertical synchronization.

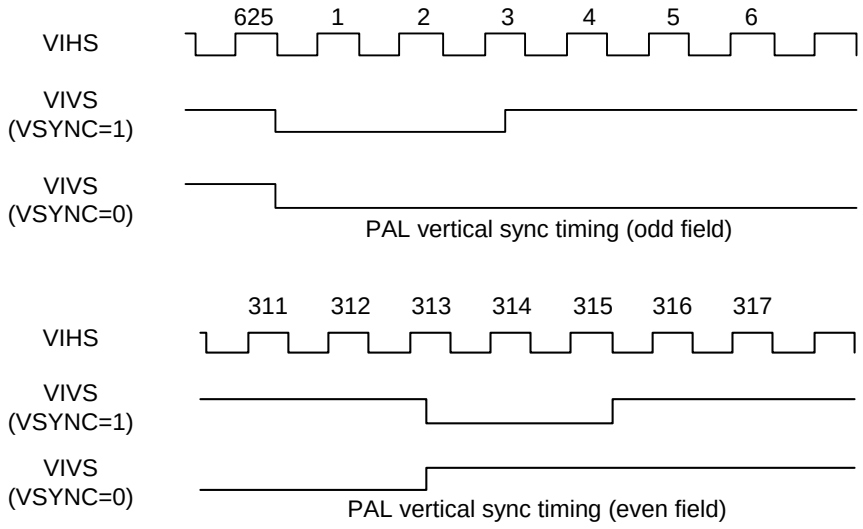
Figure 6-10 shows the NTSC vertical sync timing of the VIU (625 lines) and Figure 6-11 shows the PAL vertical sync timing of the VIU (525 lines). VI_HSYNC_VD indicates the horizontal sync pulse. When VSYNC = 1, VI_VSYNC_FIELD indicates the vertical sync pulse; when VSYNC = 0, VI_VSYNC_FIELD indicates the field sync signal.NTSC vertical sync timings





In NTSC interlaced scanning mode, the level of the vertical sync signal in field 1 becomes low at the start of line 4, remains low for three consecutive lines, and then becomes high at the start of line 7. The VIU receives 240-line data from line 22 to line 261. The level of the vertical sync signal in the second field becomes low in the middle of line 266, remains low for three consecutive lines, and then becomes high in the middle of line 269. The VIU receives 240-line data from line 285 to line 524.

Figure 6-11 PAL vertical sync timings



In PAL interlaced scanning mode, the level of the vertical sync signal in field 1 becomes low at the start of line 1, remains low for 2.5 consecutive lines, and then becomes high in the middle of line 3. The VIU receives 288-line data from line 24 to line 310. The level of the vertical sync signal in the second field becomes low in the middle of line 313, remains low for 2.5 consecutive lines, and then becomes high at the start of line 316. The VIU receives 288-line data from line 336 to line 623.

The preceding timings are typical BT.601 vertical timings. The following parameters are configurable: the number of lines from the start of the field to the valid line, the number of field valid lines, and the polarity of the vertical sync timing.

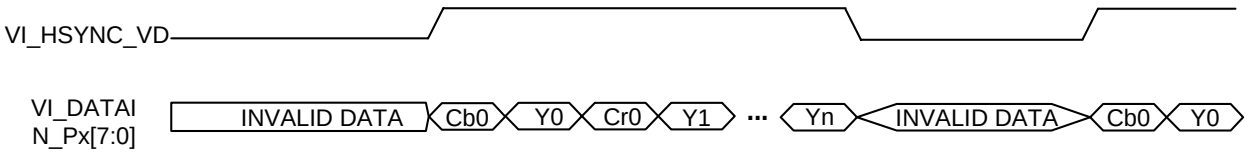
DC Interface Timings

The VIU supports the DC data transfer of up to QXGA (2048 x 1536) resolution.

1. Horizontal timing

When a DC is connected to the VIU, VI_HSYNC_VD indicates the data valid signal. The polarity of this signal is configurable. Figure 6-12 shows the horizontal timing.

Figure 6-12 DC horizontal timing



2. Vertical timing

The VIU supports the pulse vertical timing and line valid vertical timing, as shown in Figure 6-13 and Figure 6-14. In addition, the vertical sync polarity is configurable.

Figure 6-13 DC vertical pulse timing

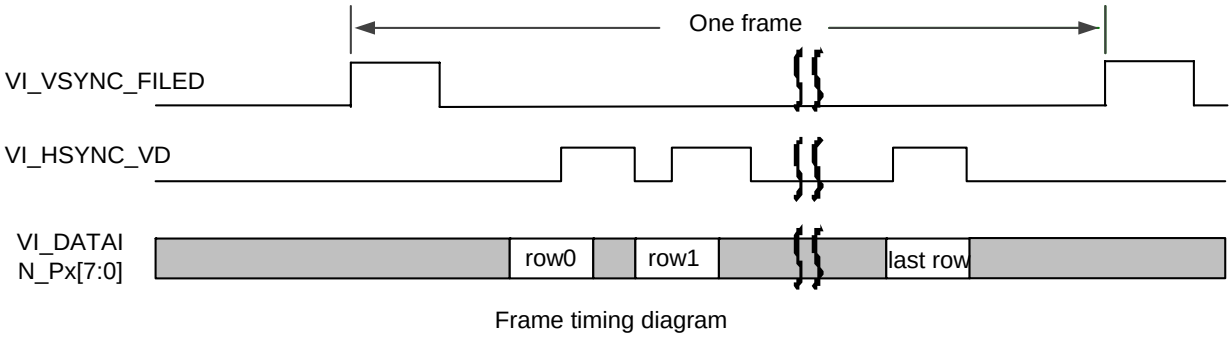
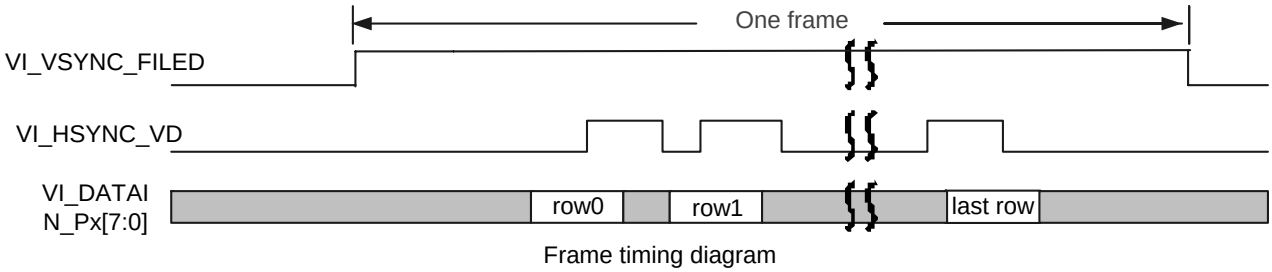


Figure 6-14 DC vertical line valid timing



The VIU processes the preceding two timings in the same way. To be specific, the VIU considers the start of a frame after it detects a rising edge or a falling edge, and then detects the data valid signal to check whether the current data is valid.

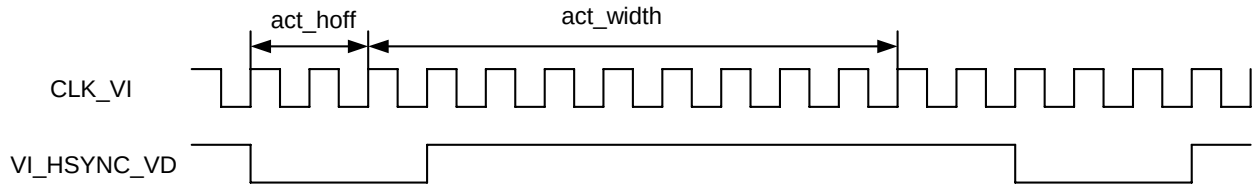
16-Bit Sync Interface

The VIU supports the 16-bit parallel interface with Y/C separate inputs and separate sync signals. Port 0 and port 1 belong to the same group and sync signals are from port 0. Port 2 and port 3 also belong to the same group and sync signals are from port 2.

1. Horizontal timing

Figure 6-15 shows the horizontal sync timing of the 16-bit sync parallel interface.

Figure 6-15 Horizontal sync timing of the 16-bit sync parallel interface

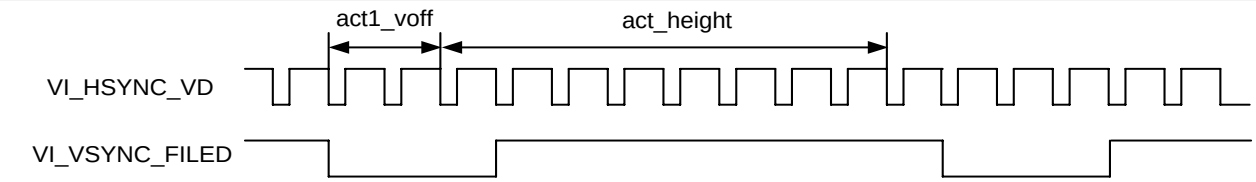


act_hoff and act_width are configurable. For details, see the description of VI_Px_HSYNC. vi_px_hsync_vd is also configurable. For details, see the description of VIn_PORT_CFG.

2. Vertical timing

Figure 6-16 shows the vertical sync timing of the 16-bit sync parallel interface.

Figure 6-16 Vertical sync timing of the 16-bit sync parallel interface

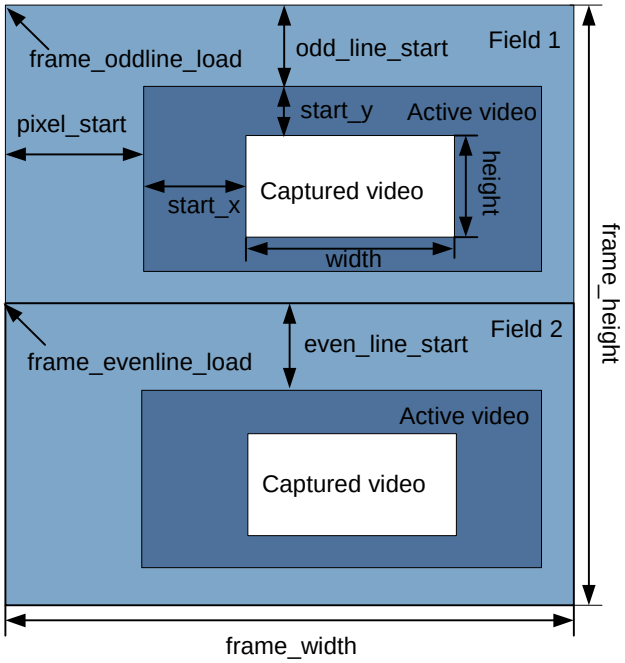


act1_voff and act1_height are configurable. For details, see the description of the register VI_Px_VSYNC1. vi_px_vsync_field is also configurable. For details, see the description of the register VIn_PORT_CFG.

Active Video Range

As shown in Figure 6-17, an active video starts from the end of the horizontal blanking area and the vertical blanking area. The actual view range, however, is within the active video range. That is, compared with the boundary of the active video, the boundary of the actual view is shrunk; therefore, the boundary effect is avoided.

Figure 6-17 Relationships between the active video area and the horizontal/vertical blanking areas



NOTE

For the definitions of related parameters, see the description of the register VIn_CAP_START.

Image Data Collection

The VIU is mainly used to collect video data streams and store the streams in the DDR. In addition, the VIU supports the chrominance conversion from the co-sited format to interspersed format in the horizontal direction and 1/2 down scaling of luminance and chrominance.

1. Two types of YCbCr 4:2:2 chrominance sampling modes

Figure 6-18 and Figure 6-19 show the horizontal chrominance relationship in co-sited and interspersed formats respectively.

Figure 6-18 YCbCr 4:2:2 co-sited sampling format

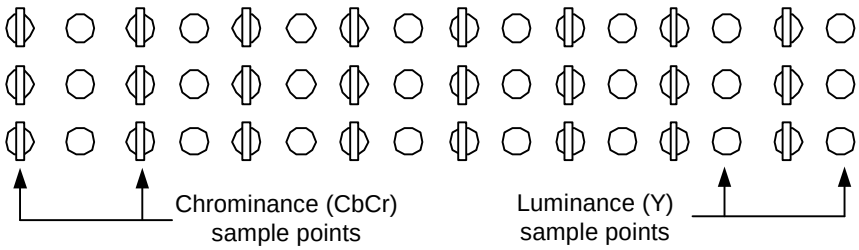
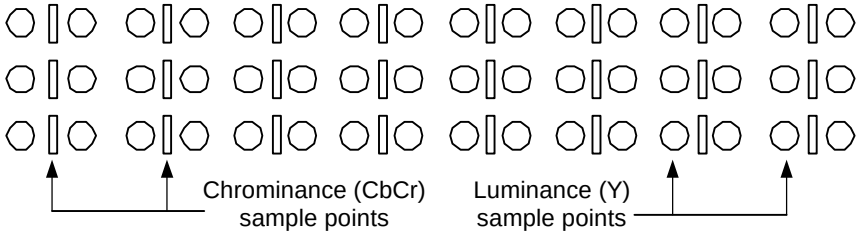


Figure 6-19 YCbCr 4:2:2 interspersed sampling format



2. Conversion from the co-sited format to the interspersed format

The inputs of the BT.656/601 and YCbCr 4:2:2 standard interface are co-sited data. The VIU supports data collection and storage in the co-sited format and data storage in the interspersed format.

- Co-sited sampling

  In this format, chrominance and luminance components of input images are written to the memory, but the sampling positions of chrominance and luminance components are not changed. Therefore, the sampling chrominance and luminance components of buffered images are the same as those in Figure 6-18. That is, the sampling format is still the YCbCr 4:2:2 co-sited format.

- Interspersed sampling

In this format, chrominance signals are resampled, which causes the sampling chrominance component to be located in the middle of the chrominance components, as shown in Figure 6-19.

3. Horizontal 1/2 down scaling

When an input image is 1/2 scaled-down in the horizontal direction, the VIU reduces the horizontal resolution (Y, Cb, and Cr) of the image by half.

- For the chrominance component (Y), 8-order FIR filter is used to carry out interpolation filtering.
- For the luminance components (Cb and Cr), 4-order FIR filter is used to carry out interpolation filtering.

During the process of horizontal 1/2 down scaling, the sampling format can be converted from the co-sited format into the interspersed format at the same time.

Image Storage Modes

The image storage modes are as follows:

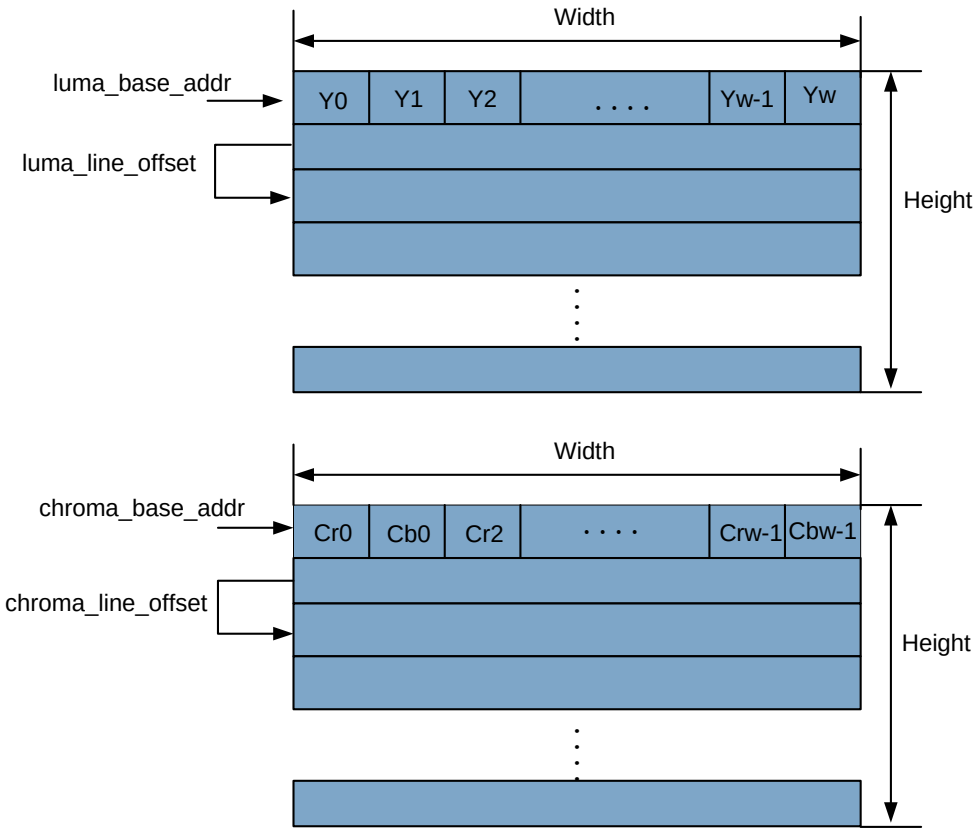
- Semi-planar YCbCr storage mode
- Package storage mode
- Raw data storage mode

1. Semi-planar YCbCr 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.

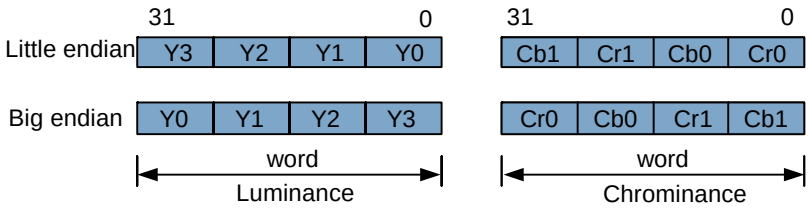
- For one line, the luminance component and chrominance component are stored continuously.
- For two continuous lines, the components are stored based on the value of the parameter offset. This parameter defines the storage stride between the start of two lines. The storage locations in the DDR of the luminance component and chrominance component depend on the start address base_addr. Figure 6-20 shows the mode of storing the YCbCr4:2:2 data captured by the VIU.

Figure 6-20 YCbCr4:2:2 storage mode



In the DDR, data is stored in the unit of word (32 bits). A 32-bit word consists of four 8-bit pixels in big endian mode or little endian mode. Figure 6-21 shows the big endian and little endian storage modes by taking the luminance component and chrominance component as examples.

Figure 6-21 Big endian and little endian storage modes

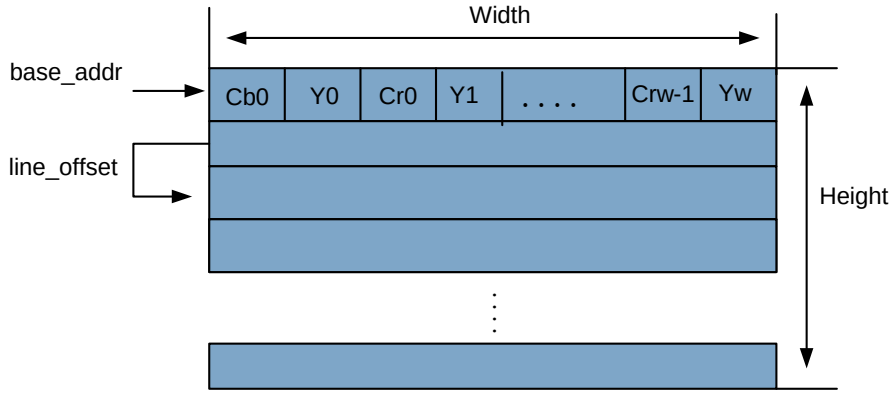


The VIU supports the DDR that stores data in little endian mode only.

2. Package storage mode

Figure 6-22 shows how to store 4:2:2 Y/Cb/Cr data in package storage mode.

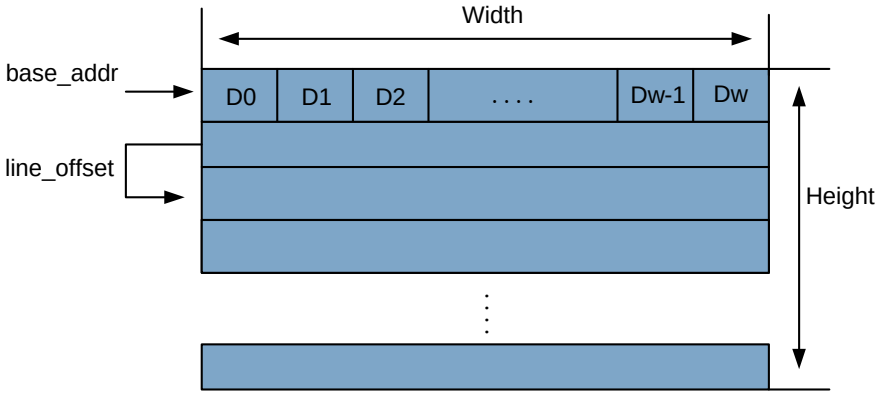
Figure 6-22 Package storage mode



3. Raw data storage mode

In raw data storage mode, data is stored in a word in sequence. In the DDR, a word consists of four 8-bit pixels. Figure 6-23 shows the storage mode of 8-bit data.

Figure 6-23 8-bit raw data storage mode



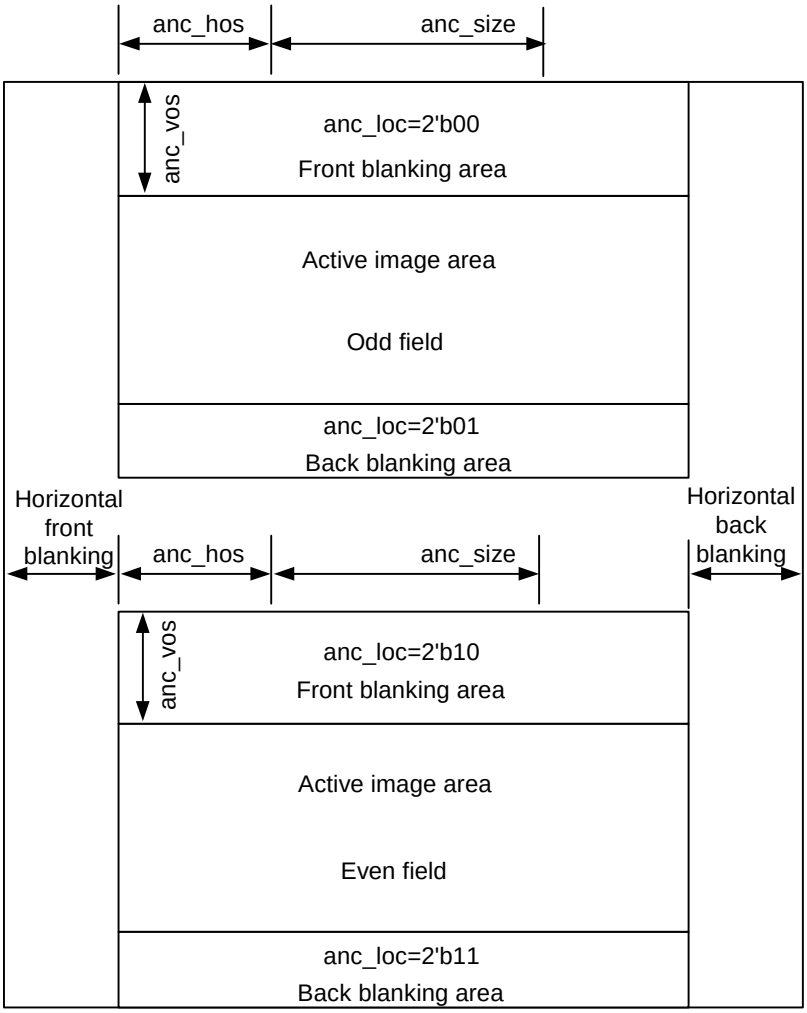
Luminance Statistics

In the VIU, the luminance statistics refers to the estimation of the average luminance of the current image, which is used to check whether the cameras are shielded. The luminance statistics is collected as follows: The input luminance components are accumulated through a 32-bit up counter. After an image is received, the accumulated value is output.

Capturing the Data of Blanking Areas

In most cases, the VIU provides data value-added services by transferring 1-line or 2-line data through blanking areas. For example, the VIU provides the function of capturing the data of blanking areas. Users need to configure the position of the data of blanking areas, as shown in Figure 6-24.

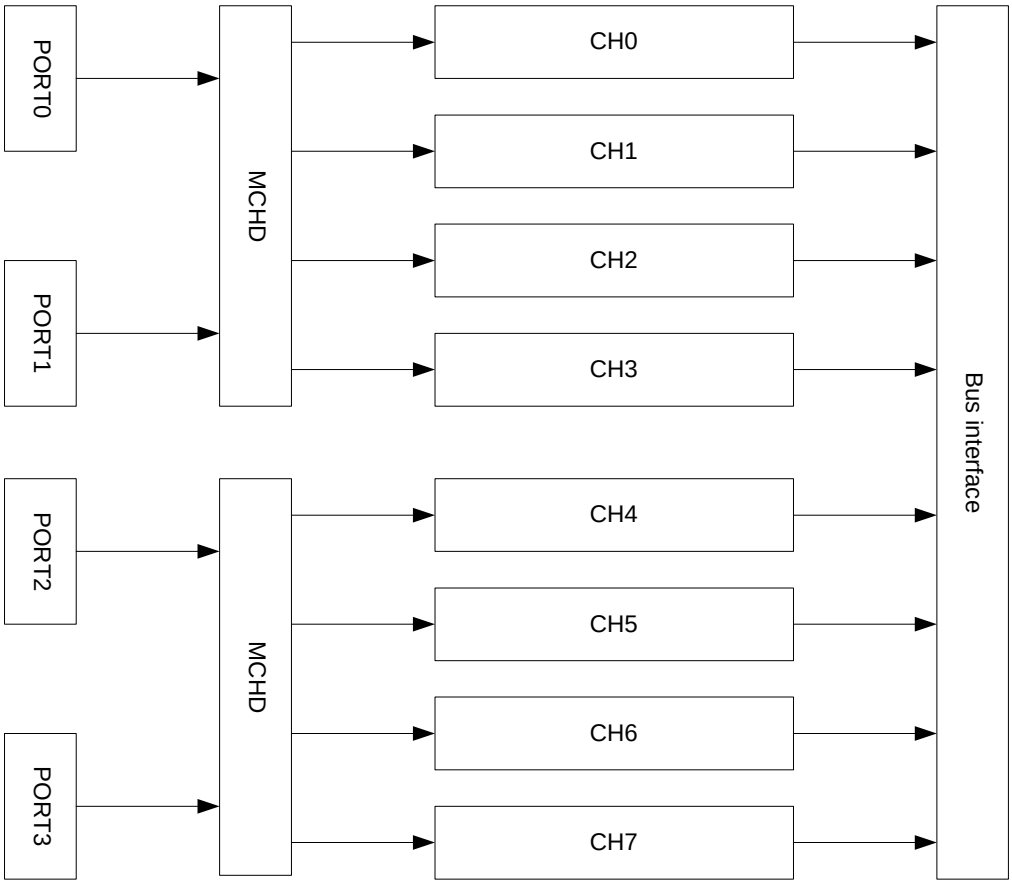
Figure 6-24 Position of the data of blanking areas



Mappings Between External Ports and Internal Channels

The VIU provides four external ports and eight internal data processing channels. The ports are used to receive data and the channels are used to process the input data of the ports. Figure 6-25 shows the relationships between the external ports and internal channels.

Figure 6-25 Mappings between the external ports and internal channels of the VIU



When the video data of ports is input in non-TDM mode, the data is processed through the uppermost channels. To be specific, the input data of port 0 is processed through channel 0; the input data of port 1 is processed through channel 2; the input data of port 2 is processed through channel 4; the input data of port 3 is processed through channel 6. When the video data is input in TDM mode, the MCHD module demultiplexes the data of ports to corresponding channels based on the TDM mode of the port and channel IDs. Port 0 and port 2 support 2-channel or 4-channel TDM video inputs; port 1 and port 3 support 2-channel TDM video inputs. When port 0 works in 4-channel TDM mode, port 1 is invalid; when port 2 works in 4-channel TDM mode, port 3 is invalid. When the data is input in Y/C separation mode, the data of port 0 is processed through channel 0, the data of port 1 is processed through channel 2, the data of port 2 is processed through channel 4, and the data of port 3 is processed through channel 6.

6.1.5 Operating Mode

6.1.5.1 Pin Multiplexing Configuration

VI pins are multiplexed with video output (VO) and general-purpose input/output (GPIO) pins. You can set the pins to VI pins by configuring the IO config registers reg0 to reg27.

6.1.5.2 Soft Reset

The soft reset of port 0 to port 3 of the VIU is controlled through SC_PERCTRL8 bit[25:22],

whereas the soft reset of the VIU bus is controlled through SC_PERCTRL8 bit[21]. For details, see the description of SC_PERCTRL8 in chapter 3 "System Controller."

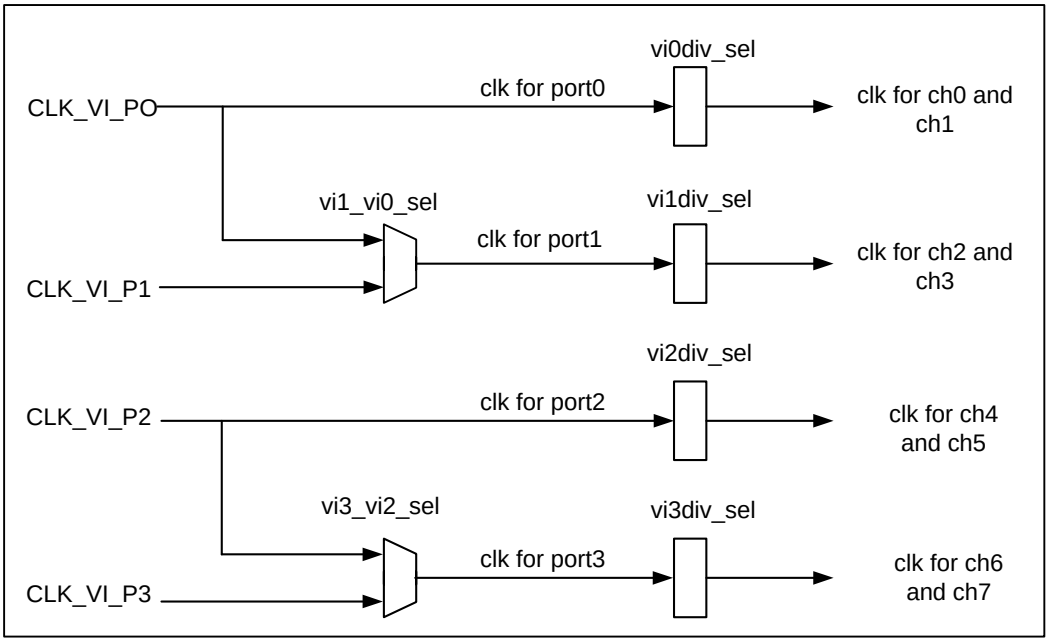
6.1.5.3 Clock Configuration

1. The VIU clocks are configured through SC_PERCTRL9 bit[13:4]. This section describes the definitions of the bits of SC_PERCTRL9 and the way of configuring the bits:

- vi1_vi0_sel and vi3_vi2_sel are port clock select bits that are used to select the clock sources of port 1 and port 3. When port 0 and port 1 work in Y/C separation input mode or port 0 works in 4-channel TDM mode, you need to set the clock source of port 1 to port 0, that is, set vi1_vi0_sel to 1. When port 2 and port 3 work in Y/C separation input mode or port 2 works in 4-channel TDM mode, you need to set the clock source of port 3 to port 2, that is, set vi3_vi2_sel to 1
- vi0div_sel, vi1div_sel, vi2div_sel, and vi3div_sel are frequency-division clock control bits. When the input mode of corresponding ports is TDM input mode, these bits are used to configure frequency dividers for internal channels. In 4-channel TDM mode, set the frequency divider to 4; in 2-channel TDM mode, set the frequency divider to 2.

2. Figure 6-26 shows the clock configuration of the VIU.

Figure 6-26 Clock configuration of the VIU



6.1.5.4 Interrupt

The interrupt ID of the VIU in the system is 17. In the VIU, each internal channel has eight interrupt sources. In addition, each channel has an interrupt enable register VIn_INT_EN and an interrupt status register VIn_INT_STATUS.

6.1.5.5 Usage Guide

Function of the reg_newer Bit of the VIU

- Before enabling a channel of the VIU through software, do as follows:

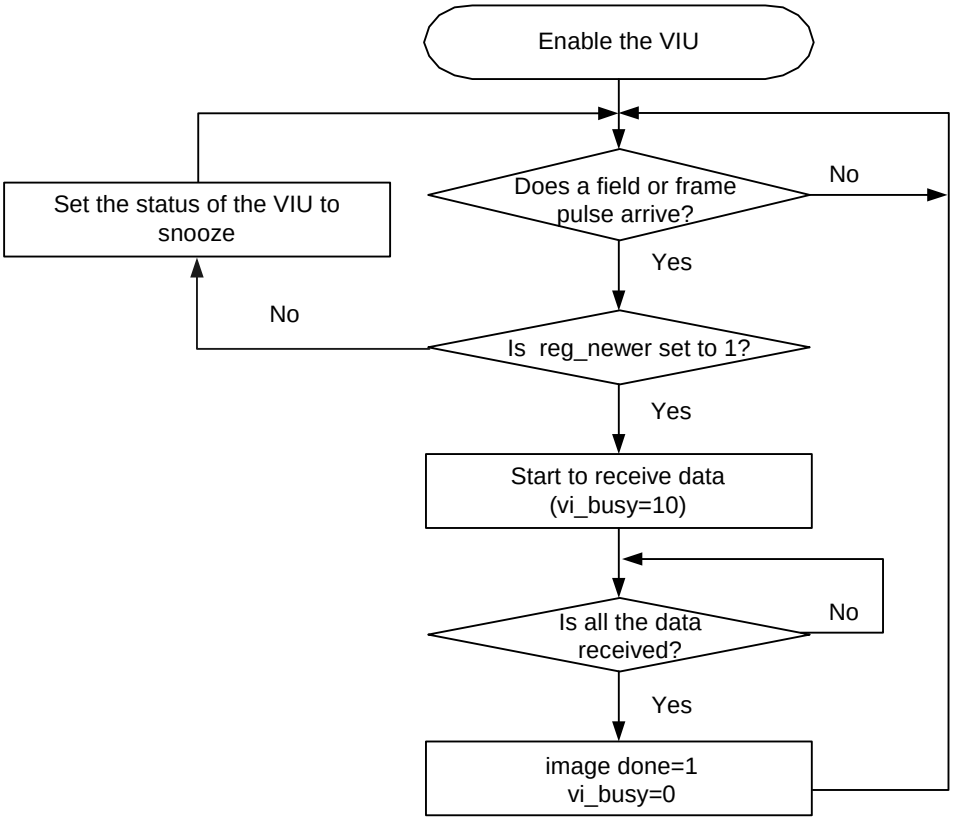
- Perform write operation on the image register VIn_REG_NEWER of the VIU.
- Write 1 to the reg_newer bit to inform the VIU that the current register is ready.

- After the VIU is enabled, the VIU logic starts to work. When a field or a frame arrives, the following cases may occur:
  - If reg_newer is 0, the VIU does not receive data, sets the hardware status to snooze, and then waits the arrival of the next field or frame.
  - If reg_newer is 1, the VIU starts to receive data. At the same time, the register update interrupt reg_update_int is generated and the hardware status is set to busy.
- After all the current data is received, the busy status of the hardware is cleared. When the next field or frame arrives, the following cases may occur:
  - If reg_newer is 0, the data of the next field or frame is dropped.
  - If reg_newer is 1, the data of the next field or frame is received.

Workflow of the VIU

Figure 6-27 shows the workflow of the VIU.

Figure 6-27 Workflow of the VIU



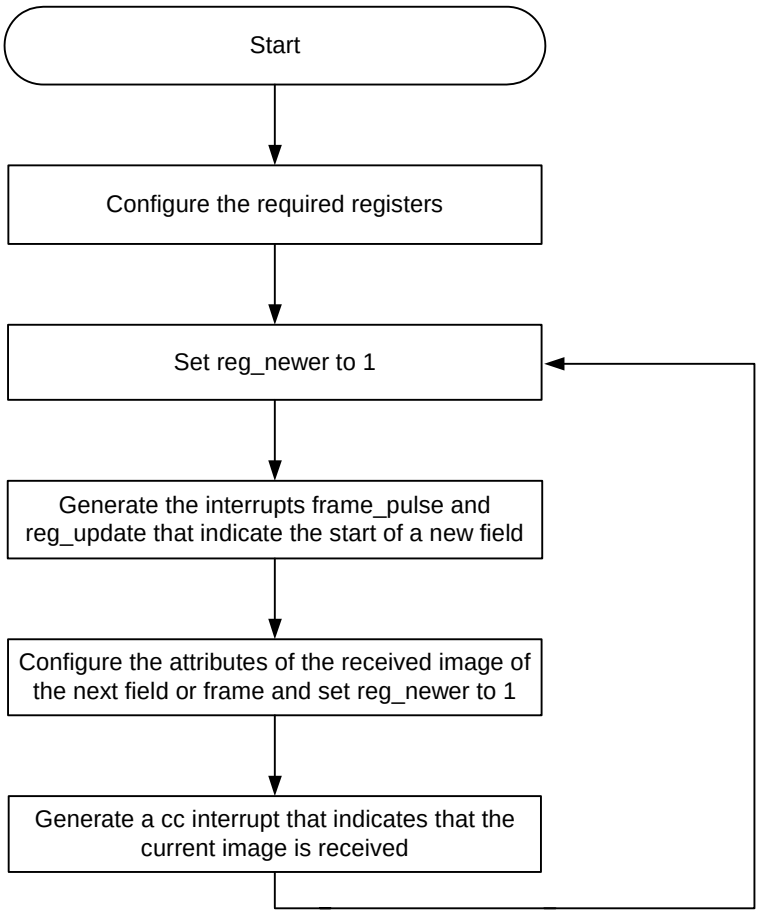
In BT.656/601 and DC modes, after the data of a specified field or frame is received, the VIU checks the reg_newer bit before the next field or frame arrives. If the reg_newer bit is 1, it indicates that software is updated or the register is a VIU confirmation register. In this case, the VIU automatically loads the register values configured through software to the working register (the software of this register is inaccessible), clears the reg_newer bit, and starts to

receive the data of the next field or frame. Otherwise, the VIU starts to receive data only when the reg_newer bit is 1 and the next field or frame arrives.

Software Configuration Process

Figure 6-28 shows the software configuration process in interrupt mode.

Figure 6-28 Software configuration process



In BT.656 and DC modes, you do not need to configure timing registers. That is, you need to configure the following registers only: VIn_CAP_SIZE, VIn_LINE_OFFSET, VIn_YBASE_ADDR, VIn_UBASE_ADDR, VIn_VBASE_ADDR, VIn_CH_CTRL, and VIn_CAP_START.

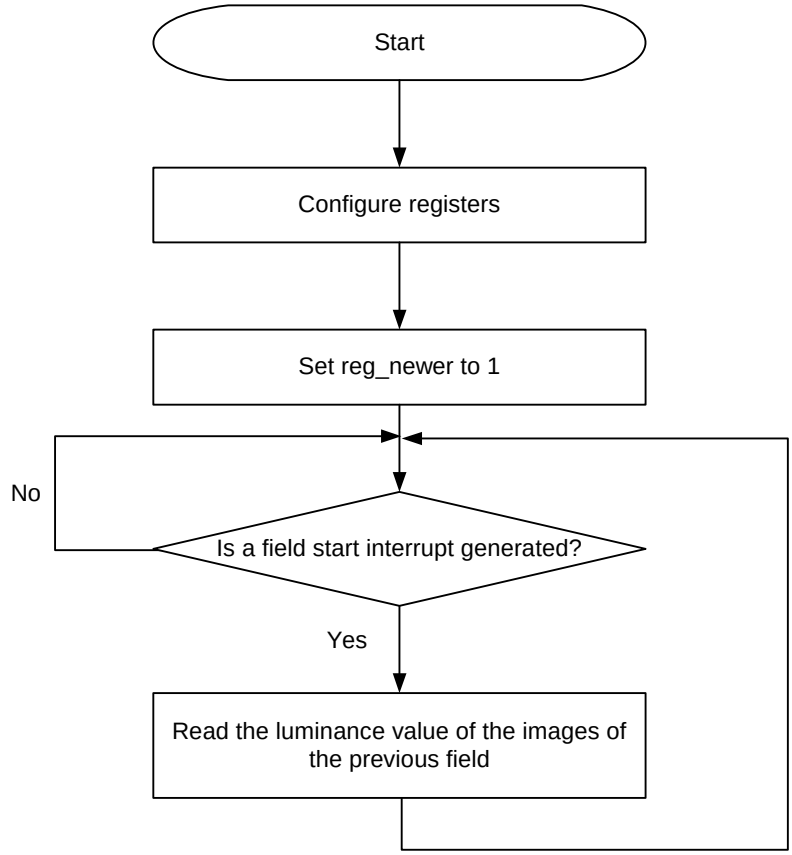
In BT.601 mode, you need to configure the timing registers VI_Px_VSYNC1, VI_Px_VSYNC2, and VI_Px_HSYNC1. For example, the values of timing registers are the default values of registers in the PAL standard.

Timing registers include vertical sync registers and horizontal sync registers.

Process of Reading the Luminance Statistical Value Through Software

Figure 6-29 shows the process of reading the luminance statistical value through software.

Figure 6-29 Process of reading the luminance statistical value through software



6.1.6 Register Summary

Table 6-7 shows the VIU registers.

Table 6-7 Summary of VIU registers (base address: 0x1010_0000)

Offset Address	Register	Description	Page
0x0000+nx0x1000	VIn_PORT_CFG	Port configuration register	6-26
0x0004+nx0x1000	VIn_CH_CFG	Channel configuration register	6-28
0x0008+nx0x1000	VIn_CH_CTRL	Channel control register	6-32
0x000C+nx0x1000	VIn_REG_NEWER	Channel register configuration completion register	6-34
0x0010+nx0x1000	VIn_CAP_START	Image capture start position Register	6-35
0x0014+nx0x1000	VIn_CAP_SIZE	Image capture size register	6-36

Offset Address	Register	Description	Page
0x0018+nx0x1000	VIn_Y_STORESIZE	Storage size register of the Y component data	6-36
0x001C+nx0x1000	VIn_U_STORESIZE	Storage size register of the Cb component data	6-37
0x0020+nx0x1000	VIn_V_STORESIZE	Storage size register of the Cr component data	6-37
0x0024+nx0x1000	VIn_LINE_OFFSET	Image storage line stride register	6-40
0x0028+nx0x1000	VIn_YBASE_ADDR	Base address register of the Y component	6-41
0x002C+nx0x1000	VIn_UBASE_ADDR	Base address register of the Cb component	6-42
0x0030+nx0x1000	VIn_VBASE_ADDR	Base address register of the Cr component	6-43
0x0034	VI_INT_DLY_CNT	Interrupt delay counter register	6-43
0x0038+nx0x1000	VIn_INT_EN	Interrupt enable register	6-43
0x003C+nx0x1000	VIn_INT_STATUS	Interrupt status register	6-45
0x0040+nx0x1000	VIn_RAW_INT	Raw interrupt status register	6-46
0x0044	VI_INT_INDICATOR	Interrupt indicator register	6-47
0x0048+nx0x1000	VI_RAW_INT_INDICATOR	Raw interrupt indicator register	6-49
0x004C+nx0x1000	VIn_STATUS	Status register	6-51
0x0050+nx0x1000	VIn_LUM_ADDER	Luminance adder register	6-52
0x0054+nx0x1000	VIn_LUM_STRH	Luminance stretch register	6-52
0x0058+nx0x1000	VIn_LUM_DIFF_ADDER	Luminance difference adder register	6-54
0x005C+nx0x1000	VIn_BLOCK0_START	Start position register of image block 0	6-55
0x0060+nx0x1000	VIn_BLOCK1_START	Start position register of image block 1	6-55
0x0064+nx0x1000	VIn_BLOCK2_START	Start position register of image block 2	6-55
0x0068+nx0x1000	VIn_BLOCK3_START	Start position register of image block 3	6-56
0x006C+nx0x1000	VIn_BLOCK0_SIZE	Block 0 size register	6-56

Offset Address	Register	Description	Page
0x0070+nx0x1000	VIn_BLOCK1_SIZE	Block 1 size register	6-57
0x0074+nx0x1000	VIn_BLOCK2_SIZE	Block 2 size register	6-57
0x0078+nx0x1000	VIn_BLOCK3_SIZE	Block 3 size register	6-57
0x007C+nx0x1000	VIn_BLOCK0_COLOR	Filling color register of image block 0	6-58
0x0080+nx0x1000	VIn_BLOCK1_COLOR	Filling color register of image block 1	6-58
0x0084+nx0x1000	VIn_BLOCK2_COLOR	Filling color register of image block 2	6-59
0x0088+nx0x1000	VIn_BLOCK3_COLOR	Filling color register of image block 3	6-59
0x009C+nx0x1000	VIn_ANC0_START	Start position register of blanking area data block 0	6-60
0x00A0+nx0x1000	VIn_ANC0_SIZE	Size register of blanking area data block 0	6-61
0x00A4+nx0x1000	VIn_ANC1_START	Start position register of blanking area data block 1	6-61
0x00A8+nx0x1000	VIn_ANC1_SIZE	Size register of blanking area data block 1	6-62
0x00AC+nx0x1000	VIn_ANC0_WORD	Blanking area data register	6-62
0x00CC+nx0x1000	VIn_ANC1_WORD	Blanking area data register	6-63
0x00EC	VI_P0_VSYNC1	Odd field vertical sync register of port 0	6-64
0x80EC	VI_P2_VSYNC1	Odd field vertical sync register of port 2	6-64
0x00F0	VI_P0_VSYNC2	Even field vertical sync register of port 0	6-65
0x80F0	VI_P2_VSYNC2	Even field vertical sync register of port 2	6-66
0x00F4	VI_P0_HSYNC	Horizontal sync register of port 0	6-66
0x80F4	VI_P2_HSYNC	Horizontal sync register of port 2	6-67
0x00F8	VI_PRIO_CONFIG	Bus priority control register.	6-67
0x00FC+nx0x1000	VIn_LUM_COEF0	Luminance filtering coefficient register 0	6-70

Offset Address	Register	Description	Page
0x0100+nx0x1000	VIn_LUM_COEF1	Luminance filtering coefficient register 1	6-71
0x0104+nx0x1000	VIn_LUM_COEF2	Luminance filtering coefficient register 2	6-71
0x0108+nx0x1000	VIn_LUM_COEF3	Luminance filtering coefficient register 3	6-72
0x010C+nx0x1000	VIn_CHROMA_COEF0	Chrominance filtering coefficient register 0	6-72
0x0110+nx0x1000	VIn_CHROMA_COEF1	Chrominance filtering coefficient register 1	6-73

NOTE

The letter n in the register VIn_ABC indicates the ID of a VI channel. In addition, n is an integer and ranges from 0 to 7.

6.1.7 Register Description

VIn_PORT_CFG

VIn_PORT_CFG is the port configuration register that is used to configure the parameters of each external port.

This register is valid only when n is 0, 2, 4, or 6.

Offset Address	Register Name	Total Reset Value
0x0000+nx0x1000	VI_PORT_CFG	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																				port_scan_mode		port_cap_mode		port_mux_mode		reserved	port_vsync	port_vsync_neg	port_hsync	port_hsync_neg	port_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:12]	RO	reserved	Reserved.
[11:10]	RW	port_scan_mode	Port data input mode. Scan_mode[1]=0: Y/C composite input mode Scan_mode[1]=1: Y/C separation input mode Scan_mode[0]=0: interlaced input mode Scan_mode[0]=1: progressive input mode
[9:8]	RW	port_cap_mode	Port data receive mode. 00: BT.656 mode 01: BT.601 mode 10: DC mode Others: reserved **Note:** When port 1 and port 3 work in non-HD mode, the port data receive mode must be BT.656 mode.
[7:6]	RW	port_mux_mode	Port multi-channel TDM input working mode. 00: 1-channel D1 data 01: 2-channel D1 TDM data 01: 4-channel D1 or half D1 TDM data Others: reserved **Note**: Port 1 and port 3 do not support the 4-channel TDM mode.
[5]	RO	reserved	Reserved.
[4]	RW	port_vsync	Configuration signal of the pin VI_P_VSYNC_FIELD. 0: field number (odd field or even field) or line valid signal • In BT.601 mode, this signal indicates the field number. • In DC mode, this signal indicates the line valid signal. 1: vertical sync pulse **Note:** You do not need to configure this bit for the 16-bit interface.

[3]	RW	port_vsync_neg	Polarity configuration signal of the pin VI_P_VSYNC_FIELD. 0: active high In pulse mode (port_vsync = 1), positive pulse indicates sync pulse. In field number mode (port_vsync = 0), the high level indicates even field and the low level indicates odd field. In line valid mode, high level indicates line valid. 1: active low In pulse mode (port_vsync = 1), negative pulse indicates sync pulse. In field number mode (port_vsync = 0), the low level indicates even field and the low level indicates odd field. In line valid mode, the low level indicates line valid.
[2]	RW	port_hsync	Configuration signal of the pin VI_P_HSYNC_VD. 0: data valid signal 1: horizontal sync pulse signal **Note:** Set this bit to 1 for the 16-bit interface.
[1]	RW	port_hsync_neg	Polarity configuration signal of the pin VI_P_HSYNC_VD. 0: active high In pulse mode (port_hsync = 1), the positive pulse indicates the sync pulse signal. In data valid mode (port_hsync = 0), the high level indicates the data valid signal. 1: active low In pulse mode (port_hsync = 1), the negative pulse indicates the sync pulse signal. In data valid mode (port_hsync = 0), the low level indicates the data valid signal.
[0]	RW	port_en	Port enable. 0: disabled 1: enabled

VIn_CH_CFG

VIn_CH_CFG is the channel configuration register that is used to configure the parameters of each external channel.

Offset Address	Register Name	Total Reset Value
0x0004+nx0x1000	VIn_CH_CFG	0x0000_1480

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					fix_code	yc_channel	seav_f_neg	chrom_swap	ch_id_en	ch_id		reserved	even_line_sel			odd_line_sel			correct_en	down_scaling	chroma_resample	store_method		cap_sel		cap_seq		reserved		store_mode	data_width	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:27]	RO	reserved	Reserved.
[26]	RW	fix_code	Configuration of the most significant bit (MSB) of the BT.656 timing reference code. 0: fixed at 1 1: fixed at 0
[25]	RW	yc_channel	Y/C configuration of the current channel. 0: The current channel is the luminance Y channel. 1: The current channel is the Chrominance C channel. **Note:** This register is valid only in Y/C separation mode.
[24]	RW	seav_f_neg	Polarity of the field indication bit (F) of the BT.656 timing reference code. 0: 1st field:F = 0 and 2nd field:F = 1 (standard) 1: 1st field:F = 1 and 2nd field:F = 0 (non-standard)
[23]	RW	chrom_swap	Channel data swap enable. 0: output in normal way 1: byte reverse in halfword **Note:** This function is available in HD Y/C separation input mode only. Channel 4 and channel 12 are used to transfer chrominance. The sequence of storing chrominance can be set as follows: 0: Cr1Cb1Cr0Cb0 1: Cb1Cr1Cb0Cr0
[22]	RW	ch_id_en	Channel ID detect enable. 0: disabled 1: enabled **Note**: This function is available only when ports work in 2-channel or 4-channel TDM mode.

[21:20]	RW	ch_id	Channel ID. 00: data with the channel ID 00 01: data with the channel ID 01 10: data with the channel ID 10 11: data with the channel ID 11 **Note**: If the channel ID detect function is enabled, a channel allows data to pass only when the channel ID in the data is the same as its channel ID.
[19]	RO	reserved	Reserved.
[18:16]	RW	even_line_sel	Line selection for collecting chrominance data of the images of even fields. 000: collect the data of odd lines only 001 collect the data of even lines only 010 collect the data of both odd and even lines 011: collect the data of line 0 and line 3 every four line. This data collection method is used only in field storage mode (store_mode = 0). 100 collect the data of line 1 and line 2 every four line. This data collection method is used only in field storage mode (store_mode = 0). **Note**: The line number starts from 0. The interlaced capture function is used to obtain YCbCr 4:2:0 data, but is available for the chrominance component only. The luminance data of both odd and even lines is collected. In frame input mode (DC mode and progressive HD mode), even_line_sel controls whether to drop chrominance lines. In frame storage mode, it is not allowed to configure the collect mode to collect odd-line data only in odd fields or collect even-line data only in even fields. It is recommended to configure the collect mode to collect even-line data only in odd field or collect odd-line data only in odd fields. In HD mode, only channel 2 or channel 6 needs to be configured.
[15:13]	RW	odd_line_sel	Line selection for collecting chrominance data of the images of odd fields. 000: collect the data of odd lines only 001 collect the data of even lines only 010 collect the data of both odd and even lines 011: collect the data of line 0 and line 3 every four line. This data collection method is used only in field storage mode (store_mode = 0). 100 collect the data of line 1 and line 2 every four line. This data collection method is used only in field storage mode (store_mode = 0). **Note**: The line number starts from 0. The interlaced capture function is used to obtain YCbCr 4:2:0 data, but is available for the chrominance component only. The luminance data of both odd and even lines is collected. In frame input mode (DC mode and progressive HD mode), odd_line_sel is invalid. In frame storage

			mode, it is not allowed to configure the collect mode to collect odd-line data only in odd fields or collect even-line data only in even fields. It is recommended to configure the collect mode to collect even-line data only in odd field or collect odd-line data only in odd fields. In HD mode, only channel 2 or channel 6 needs to be configured.
[12]	RW	correct_en	SAV and EAV data check enable. 0: disabled 1: enabled
[11]	RW	down_scaling	Horizontal 1/2 down scaling. 0: disabled 1: enabled **Note**: This function is not supported in HD Y/C separation input mode.
[10]	RW	chroma_resample	Chrominance resampling. 0: disabled 1: enabled The sampling format is converted from co-sited into interspersed. **Note**: During the process of chrominance resampling, the sampling format can be converted from co-sited into interspersed only. In addition, this function is not supported in HD Y/C separation input mode.
[9:8]	RW	store_method	Storage method. 00: planar Y/Cb/Cr 01: semi-planar YCbCr 10: package YCbCr4:2:2 11: raw data, for HD Y/C separation input mode only
[7:6]	RW	cap_sel	Field selection for collection image data. 00: collect the data of odd fields (top fields) only 01: collect the data of even fields (bottom fields) only 10: collect the data of both odd and even fields Others: reserved
[5:4]	RW	cap_seq	YCbCr input sequence bit. 00: CbYCrY 01: CrYCbY 10: YCbYCr 11: YCrYCb
[3]	RO	reserved	Reserved.
[2]	RW	store_mode	Storage mode. 0: field storage mode 1: frame storage mode **Note**: If ports work in progressive input mode, the storage mode of

			this register must be set to field storage mode.
[1:0]	RW	data_width	Data width. 00: 8 bits 10: 10 bits Others: reserved

VIn_CH_CTRL

VIn_CH_CTRL is the control register that controls the start time and end time of capturing data. In addition, VIn_CH_CTRL can enable the VIU and control the reg_newer interrupt.

Offset Address	Register Name	Total Reset Value
0x0008+nx0x1000	VIn_CH_CTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																debug_en	lum_strh_en	anc1_en	anc0_en	reserved				block3_en	block2_en	block1_en	block0_en	reserved			ch_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15]	RW	debug_en	Debug mode enable. 0: disabled 1: enabled If the debug mode is not enabled, the value 0 is returned when debug registers are read.
[14]	RW	lum_strh_en	Luminance stretch enable. 0: do not stretch the luminance 1: stretch the luminance **Note**: The luminance stretch function is not supported in Y/C separation input mode.
[13]	RW	anc1_en	Capture enable of data block 1 in the blanking area. 0: do not capture 1: capture
[12]	RW	anc0_en	Capture enable of data block 0 in the blanking area. 0: do not capture 1: capture
[11:8]	RO	reserved	Reserved.
[7]	RW	block3_en	Block 3 mask enable. 0: disabled 1: enabled
[6]	RW	block2_en	Block 2 mask enable. 0: disabled 1: enabled
[5]	RW	block1_en	Block 1 mask enable. 0: disabled 1: enabled

[4]	RW	blcok0_en	Block 0 mask enable. 0: disabled 1: enabled
[3:1]	RO	reserved	Reserved.
[0]	RW	ch_en	VIU channel enable. 0: disabled 1: enabled **Note**: To switch the mode (such as multi-channel switch or the switch from the BT.656 mode to the BT.601 mode), you must set this bit to 0 before switching and set this bit to 1 after switching.

VIn_REG_NEWER

VIn_REG_NEWER is the channel register configuration completion register. It is used to notify the channel whether the register for receiving the next field or frame is configured.

Offset Address	Register Name	Total Reset Value
0x000C+nx0x1000	VIn_REG_NEWER	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RW	reg_newer	Indicates whether the register for receiving the next field or frame is ready. 0: The register is not ready, so the hardware drops the next field or frame. 1: The register is ready, so the hardware starts to receive the next field or frame when the hardware detects the start of the next field or frame. **Note**: The VIU hardware clears this bit automatically after updating its internal registers.

VIn_CAP_START

VIn_CAP_START is the image capture start position register that is used to configure the start position of capturing an image.

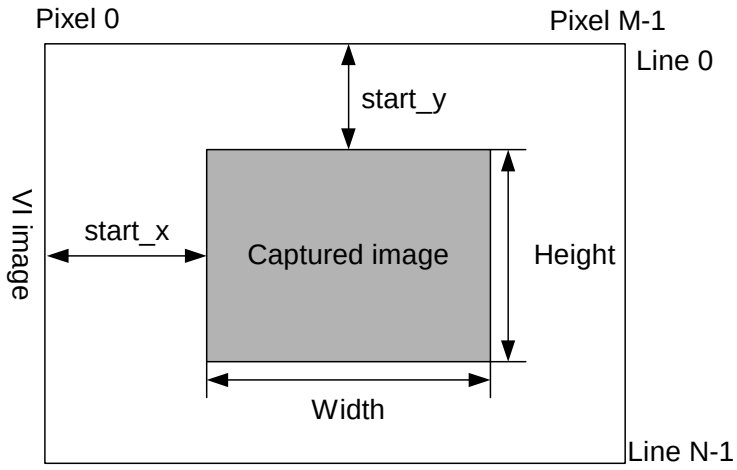
VIn_CAP_SIZE is the image capture size register that is used to configure the size of a captured image.

The registers VIn_CAP_START and VIn_CAP_SIZE show how to capture a rectangle image from a VI image, as shown in Figure 6-30.

- VIn_CAP_START describes the start coordinates start_x and start_y of the captured image from the VI image.
- VIn_CAP_SIZE describes the size, width, and height of the captured image.

Width and start_x are in the unit of input pixel (namely, luminance pixel). When the captured image is in the YUV4:2:2 format, height and start_y are in the unit of line.

Figure 6-30 Parameter diagram of the captured image



NOTE

- start_x and start_y occupy 12 bits respectively in VIn_CAP_START.
- Width and height occupy 12 bits respectively in VIn_CAP_SIZE.

Offset Address	Register Name	Total Reset Value
0x0010+nx0x1000	VIn_CAP_START	0x0000_0000

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

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	start_y	Number of the line from which images start to be captured.

[11:0]	RW	start_x	Number of the pixel from which images start to be captured.

VIn_CAP_SIZE

VIn_CAP_SIZE is the image capture size register.

Offset Address	Register Name	Total Reset Value
0x0014+nx0x1000	VIn_CAP_SIZE	0x0012_02d0

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

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	height	Height of a captured image (in lines). In frame storage mode, the height of a captured image is half of the actual height of the frame image. The VIU captures two fields to form a frame. The minimum height of a captured image supported by the VIU is one line.
[11:0]	RW	width	Width of a line of a captured image (in pixels). The minimum width of a capture image supported by the VIU is four pixels.

VIn_Y_STORESIZE

VI_Y_STORESIZE is the storage size register of the Y component data.

Offset Address	Register Name	Total Reset Value
0x0018+nx1000	VIn_Y_STORESIZE	0x000F_00A0

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

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	y_height	Height of storing the Y component data (in lines). **Note**: In Y/C separation mode, if the chrominance channel outputs YCbCr4:2:0 data, the chrominance height must be set to half of the luminance height.

[11:0]	RW	y_width	Value of subtracting 1 from the width of storing the Y component data. The line width is on the basis of 128 bits. If the line tail is less than 128 bits, the value of 128 bits is considered.

VIn_U_STORESIZE

VIn_U_STORESIZE is the storage size register of the Cb component data. In package storage mode, this register is unavailable. In the semi-planar YCbCr data format, this register indicates the chrominance storage size; in the planar YCbCr data format, this register indicates the Cb component storage size.

Offset Address	Register Name	Total Reset Value
0x001C+nx0x1000	VIn_U_STORESIZE	0x000F_00A0

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

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	c_height	In semi-planar YCbCr mode, c_height indicates the height of storing the chrominance component data (in lines). In planar YCbCr mode, c_height indicates the height of storing the Cb component data (in lines). c_height is invalid in package storage mode and raw data receive mode.
[11:0]	RW	c_width	In semi-planar YCbCr mode, c_width indicates the value of subtracting 1 from the width of storing the Y component data. The line width is on the basis of 128 bits. If the line tail is less than 128 bits, the value of 128 bits is considered. In planar YCbCr mode, c_width indicates the width of storing the Cb component data. The width is in the unit of word. If the width is less than a word, a complete word is considered. c_width is invalid in package storage mode and raw data receive mode.

VIn_V_STORESIZE

VIn_V_STORESIZE is the storage size register of the Cr component data.

Offset Address	Register Name	Total Reset Value
0x0020+nx0x1000	VIn_V_STORESIZE	0x000F00A0

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

Name	reserved	v_height	v_width
Reset	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0		

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	v_height	Height of storing the Cr component data (in lines).
[11:0]	RW	v_width	Value of subtracting 1 from the width of storing the Cr component data. The line width is on the basis of 128 bits. If the line tail is less than 128 bits, the value of 128 bits is considered. v_width is invalid in semi-planar YCbCr storage mode, package YCbCr storage mode, and raw data receive mode.

To calculate the size of the stored data when YCbCr4:2:2 data is being received, do as follows:

Step 1 Check the parity of cap_width. If cap_width is an even, the data size keeps unchanged; if cap_width is an odd, the data size must be subtracted from 1.

```
cap_width=cap_width-cap_width%2
```

Step 2 Check whether 1/2 down scaling is performed.

```
down_scaling?
yes:cap_width=cap_width/2
no:cap_width=cap_width
```

Step 3 Calculate the sizes of the stored data in different storage modes.

1. Y/Cb/Cr planar storage mode

   The data bit width is 8 bits.

```
y_width=if(cap_width%16==0)
                cap_width/16
           else
                cap_width/16+1
c_width = if(cap_width%32==0)
                cap_width/32
           else
                cap_width/32+1
v_width=if(cap_width%32==0)
                cap_width/32
           else
                cap_width/32+1
```

   The data bit width is 10 bits.

```
y_width=if(cap_width%8==0)
                cap_width/8
           else
                cap_width/8+1
c_width=if(cap_width%16==0)
```

```
                cap_width/16
            else
                cap_width/16+1
v_width=if(cap_width%16==0)
                cap_width/16
            else
                cap_width/16+1
```

2. Planar Y/C storage mode

The data bit width is 8 bits.

```
y_width=if(cap_width%16==0)
                cap_width/16
            else
                cap_width/16+1
c_width=if(cap_width%16==0)
                cap_width/16
            else
                cap_width/16+1
```

The data bit width is 10 bits.

```
y_width=if(cap_width%8==0)
                cap_width/8
            else
                cap_width/8+1
c_width=if(cap_width%8==0)
                cap_width/8
            else
                cap_width/8+1
```

3. Package storage mode

The data bit width is 8 bits.

```
y_width=if(cap_width%8==0)
                cap_width/8
            else
                cap_width/8+1
```

The data bit width is 10 bits.

```
y_width=if(cap_width%4==0)
                cap_width/4
            else
                cap_width/4+1
```

4. Raw data storage mode

The data bit width is 8 bits.

```
y_width=if(cap_width%16==0)
                cap_width/16
            else
                cap_width/16+1
```

The data bit width is 10 bits.

```
y_width=if(cap_width%8==0)
            cap_width/8
        else
            cap_width/8+1
```

Step 4 Subtract 1 from the preceding values.

```
y_width=y_width-1
c_width=c_width-1
v_width=v_width-1
```

----End

VIn_LINE_OFFSET

VIn_LINE_OFFSET is the image storage line stride register that is used to configure the stride of data storage lines. This stride is in the unit of word and the register of each channel is controlled separately.

Offset Address	Register Name	Total Reset Value
0x0024+nx0x1000	VIn_LINE_OFFSET	0x0000_0000

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

Bits	Access	Name	Description
[31:20]	RW	yline_offset	The stride is on the basis of 128 bits, that is, the actual stride is the value obtained after yline_offset is moved four bits left. • In raw data storage mode, only this stride is valid. • In package YCbCr4:2:2 storage mode, only this stride is valid. • In planar Y/C or Y/Cb/Cr storage mode, yline_offset indicates the line stride of the Y component.
[19:10]	RW	uline_offset	The stride is on the basis of 128 bits, that is, the actual stride is the value obtained after uline_offset is moved four bits left. • In raw data storage mode, uline_offset is invalid. • In package YCbCr4:2:2 storage mode, uline_offset is invalid. • In planar Y/C storage mode, uline_offset indicates the line stride of the C component. • In planar Y/Cb/Cr storage mode, uline_offset indicates the line stride of the Cb component.
[9:0]	RW	vline_offset	The stride is on the basis of 128 bits, that is, the actual stride is the value obtained after vline_offset is moved four bits left. • In raw data storage mode, vline_offset is invalid. • In package YCbCr4:2:2 storage mode, vline_offset is invalid. • In planar Y/C storage mode, vline_offset is invalid.

			• In planar Y/Cb/Cr storage mode, vline_offset indicates the line stride of the Cr component.

VIn_YBASE_ADDR

VIn_YBASE_ADDR0 is the base address 0 register of the Y component. This register is used to configure the start address for storing the Y component.
The last three bits of this register are set to 0s.

Offset Address	Register Name	Total Reset Value
0x0028+nx0x1000	VIn_YBASE_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vi_ybase_addr	Base address of the Y channel. The four least significant bits (LSBs) are set to 0s and the base address is 128-bit aligned. • In raw data storage mode, only this base address is valid. • In package YCbCr4:2:2 storage mode, only this base address is valid. • In planar Y/C or Y/Cb/Cr storage mode, vi_ybase_addr indicates the header address of the Y component.

VIn_UBASE_ADDR

VIn_UBASE_ADDR is the base address 0 register of the Cb component. This register is used to configure the start address for storing the Cb component. The last three bits of this register are set to 0s.

Offset Address	Register Name	Total Reset Value
0x002C+nx0x1000	VIn_UBASE_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vi_ubase_addr	Base address of the Cb channel. The four LSBs are set to 0s and the base address is 128-bit aligned. • In raw data storage mode, vi_ubase_addr is invalid. • In package YCbCr4:2:2 storage mode, vi_ubase_addr is invalid. • In planar Y/C storage mode, vi_ubase_addr indicates the header address of the Cr component. • In planar Y/C or Y/Cb/Cr storage mode, vi_ubase_addr indicates the header address of the Cb component.

VIn_VBASE_ADDR

VIn_VBASE_ADDR0 is the base address 0 register of the Cr component. This register is used to configure the start address for storing the Cr component. The last three bits of this register are set to 0s.

Offset Address	Register Name	Total Reset Value
0x0030+nx0x1000	VIn_VBASE_ADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vi_vbase_addr	Base address of the Cr channel. The four LSBs are set to 0s and the base address is 128-bit aligned. • In raw data storage mode, vi_vbase_addr is invalid. • In package YCbCr4:2:2 storage mode, vi_vbase_addr is invalid. • In planar Y/C storage mode, vi_vbase_addr is invalid. • In planar Y/Cb/Cr storage mode, vi_vbase_addr indicates the header address of the Cr component.

VI_INT_DLY_CNT

VI_INT_DLY_CNT is the interrupt delay counter register.

Offset Address	Register Name	Total Reset Value
0x0034	VI_INT_DLY_CNT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	int_dly_cnt	Interrupt delay counter. The counter counts bus clocks. If the counted value reaches the value of this register and there are interrupts in any of the 16 channel, the VIU reports an interrupt.

VIn_INT_EN

VIn_INT_EN is the interrupt enable register.

Offset Address	Register Name	Total Reset Value
0x0038+nx0x1000	VIn_INT_EN	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																							chdiv_err_int_en	ntsc_pal_trans_int_en	frame_pulse_int_en	reg_update_int_en	proc_err_int_en	err_int_en	field_throw_int_en	buf_ovf_int_en	cc_int_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:9]	RO	reserved	Reserved.
[8]	RW	chdiv_err_int_en	Channel allocation error indicator interrupt enable. 0: disabled 1: enabled
[7]	RW	ntsc_pal_trans_int_en	Standard conversion interrupt enable. 0: disabled 1: enabled
[6]	RW	frame_pulse_int_en	Field start interrupt enable. 0: disabled 1: enabled
[5]	RW	reg_update_int_en	Register update interrupt enable. 0: disabled 1: enabled
[4]	RW	proc_err_int_en	Protection bit error interrupt enable in BT.656 mode. 0: disabled 1: enabled
[3]	RW	err_int_en	BUS error interrupt enable. 0: disabled 1: enabled
[2]	RW	field_throw_int_en	Field or frame loss interrupt enable. 0: disabled 1: enabled
[1]	RW	buf_ovf_int_en	Internal FIFO overflow interrupt error interrupt enable. 0: disabled 1: enabled
[0]	RW	cc_int_en	Data capture completion interrupt enable.

			0: field image capture completion interrupt 1: frame image capture completion interrupt

VIn_INT_STATUS

VIn_INT_STATUS is the interrupt status register. When the value 1 is written to the corresponding bit of this register, the interrupt bit is cleared. The register of each channel is controlled separately and can be masked.

Offset Address	Register Name	Total Reset Value
0x003C+nx0x1000	VIn_INT_STATUS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																							chdiv_err_int	ntsc_pal_trans_int	frame_pulse_int	reg_update_int	proc_err_int	error_int	field_throw_int	buf_ovf_int	cc_int
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:9]	RO	reserved	Reserved.
[8]	WC	chdiv_err_int	Channel allocation error indicator interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	WC	ntsc_pal_trans_int	Standard conversion interrupt status. 0: No interrupt is generated. 1: An interrupt is generated. **Note**: The standard conversion interrupt is reported if the standard is changed. If the number of valid data lines is changed, the standard conversion is considered. That is, the standard conversion is not limited to the conversion of NTSC and PAL standards.
[6]	WC	frame_pulse_int	Field start interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[5]	WC	reg_update_int	Working register update interrupt status. 0: No interrupt is generated. 1: An interrupt is generated. When VIn_CH_CFG[store_mode] is 1, this bit indicates the frame image capture completion interrupt. When VIn_CH_CFG[store_mode] is 0, this bit indicates the field image capture completion interrupt.

[4]	WC	proc_err_int	Protection bit error interrupt status (in BT.656 mode). 0: No interrupt is generated. 1: An interrupt is generated.
[3]	WC	error_int	AHB bus error interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[2]	WC	field_throw_int	Field data loss interrupt status. 0: No interrupt is generated. 1: An interrupt is generated. **Note**: If the capture mode is set to odd field only or even field only, hardware does not report the field data loss interrupt.
[1]	WC	buf_ovf_int	Internal buffer FIFO overflow interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	WC	cc_int	Current image capture completion interrupt status. 0: No interrupt is generated. 1: An interrupt is generated. When VIn_CH_CFG[store_mode] is 0, this bit indicates the field image capture completion interrupt. When VIn_CH_CFG[store_mode] is 1, this bit indicates the frame image capture completion interrupt.

VIn_RAW_INT

VIn_RAW_INT is the raw interrupt status register. If the value 1 is written to the corresponding bit of this register, its interrupt bit is cleared. The register of each channel is controlled separately. In addition, the registers are for debugging and cannot be masked.

Offset Address	Register Name	Total Reset Value
0x0040+nx0x1000	VIn_RAW_INT	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																							chdiv_err_raw_int	ntsc_pal_trans_raw_int	frame_pulse_raw_int	reg_update_raw_int	proc_err_raw_int	error_raw_int	field_throw_raw_int	buf_ovf_raw_int	cc_raw_int
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:9]	RO	reserved	Reserved.

[8]	RO	chdiv_err_raw_int	Channel allocation error indicator raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	ntsc_pal_trans_raw_int	Standard conversion raw interrupt status. 0: No raw interrupt is generated. 1: A raw interrupt is generated. **Note**: The standard conversion interrupt is reported if the standard is changed. If the number of valid data lines is changed, the standard conversion is considered. That is, the standard conversion is not limited to the conversion of NTSC and PAL standards.
[6]	RO	frame_pulse_raw_int	Field start raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	reg_update_raw_int	Working register update raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	proc_err_raw_int	Protection bit error raw interrupt status (in BT.656 mode). 0: No interrupt is generated. 1: An interrupt is generated.
[3]	RO	error_raw_int	AHB bus error raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	field_throw_raw_int	Field data loss raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	buf_ovf_raw_int	Internal buffer FIFO overflow raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	cc_raw_int	Current image capture completion raw interrupt status. 0: field image capture completion raw interrupt 1: frame image capture completion raw interrupt When store_mode is 0, this bit indicates the field image capture completion interrupt. When store_mode is 1, this bit indicates the frame image capture completion interrupt.

VI_INT_INDICATOR

VI_INT_INDICATOR is the interrupt indicator register. This register indicates the channels (16 channels) in which interrupts are generated.

Offset Address	Register Name	Total Reset Value
0x0044+nx0x1000	VI_INT_INDICATOR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																ch15_int_indicator	ch14_int_indicator	ch13_int_indicator	ch12_int_indicator	ch11_int_indicator	ch10_int_indicator	ch9_int_indicator	ch8_int_indicator	ch7_int_indicator	ch6_int_indicator	ch5_int_indicator	ch4_int_indicator	ch3_int_indicator	ch2_int_indicator	ch1_int_indicator	ch0_int_indicator
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15]	RO	ch15_int_indicator	Interrupt indicator bit of channel 15. 0: No interrupt is generated. 1: An interrupt is generated.
[14]	RO	ch14_int_indicator	Interrupt indicator bit of channel 14. 0: No interrupt is generated. 1: An interrupt is generated.
[13]	RO	ch13_int_indicator	Interrupt indicator bit of channel 13. 0: No interrupt is generated. 1: An interrupt is generated.
[12]	RO	ch12_int_indicator	Interrupt indicator bit of channel 12. 0: No interrupt is generated. 1: An interrupt is generated.
[11]	RO	ch11_int_indicator	Interrupt indicator bit of channel 11. 0: No interrupt is generated. 1: An interrupt is generated.
[10]	RO	ch10_int_indicator	Interrupt indicator bit of channel 10. 0: No interrupt is generated. 1: An interrupt is generated.
[9]	RO	ch9_int_indicator	Interrupt indicator bit of channel 9. 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	ch8_int_indicator	Interrupt indicator bit of channel 8. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	ch7_int_indicator	Interrupt indicator bit of channel 7.

			0: No interrupt is generated. 1: An interrupt is generated.
[6]	RO	ch6_int_indicator	Interrupt indicator bit of channel 6. 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	ch5_int_indicator	Interrupt indicator bit of channel 5. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	ch4_int_indicator	Interrupt indicator bit of channel 4. 0: No interrupt is generated. 1: An interrupt is generated.
[3]	RO	ch3_int_indicator	Interrupt indicator bit of channel 3. 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	ch2_int_indicator	Interrupt indicator bit of channel 2. 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	ch1_int_indicator	Interrupt indicator bit of channel 1. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	ch0_int_indicator	Interrupt indicator bit of channel 0. 0: No interrupt is generated. 1: An interrupt is generated.

VI_RAW_INT_INDICATOR

VI_RAW_INT_INDICATOR is the raw interrupt indicator register. This register indicates the channels (16 channels) in which raw interrupts are generated.

Offset Address	Register Name	Total Reset Value
0x0048+nx0x1000	VI_RAW_INT_INDICATOR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																ch15_raw_int_indicator	ch14_raw_int_indicator	ch13_raw_int_indicator	ch12_raw_int_indicator	ch11_raw_int_indicator	ch10_raw_int_indicator	ch9_raw_int_indicator	ch8_raw_int_indicator	ch7_raw_int_indicator	ch6_raw_int_indicator	ch5_raw_int_indicator	ch4_raw_int_indicator	ch3_raw_int_indicator	ch2_raw_int_indicator	ch1_raw_int_indicator	ch0_raw_int_indicator

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15]	RO	ch15_raw_int_indicator	Interrupt indicator bit of channel 15. 0: No interrupt is generated. 1: An interrupt is generated.
[14]	RO	ch14_raw_int_indicator	Interrupt indicator bit of channel 14. 0: No interrupt is generated. 1: An interrupt is generated.
[13]	RO	ch13_raw_int_indicator	Interrupt indicator bit of channel 13. 0: No interrupt is generated. 1: An interrupt is generated.
[12]	RO	ch12_raw_int_indicator	Interrupt indicator bit of channel 12. 0: No interrupt is generated. 1: An interrupt is generated.
[11]	RO	ch11_raw_int_indicator	Interrupt indicator bit of channel 11. 0: No interrupt is generated. 1: An interrupt is generated.
[10]	RO	ch10_raw_int_indicator	Interrupt indicator bit of channel 10. 0: No interrupt is generated. 1: An interrupt is generated.
[9]	RO	ch9_raw_int_indicator	Interrupt indicator bit of channel 9. 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	ch8_raw_int_indicator	Interrupt indicator bit of channel 8. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	ch7_raw_int_indicator	Interrupt indicator bit of channel 7. 0: No interrupt is generated. 1: An interrupt is generated.
[6]	RO	ch6_raw_int_indicator	Interrupt indicator bit of channel 6. 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	ch5_raw_int_indicator	Interrupt indicator bit of channel 5. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	ch4_raw_int_indica	Interrupt indicator bit of channel 4.

		tor	0: No interrupt is generated. 1: An interrupt is generated.
[3]	RO	ch3_raw_int_indicator	Interrupt indicator bit of channel 3. 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	ch2_raw_int_indicator	Interrupt indicator bit of channel 2. 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	ch1_raw_int_indicator	Interrupt indicator bit of channel 1. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	ch0_raw_int_indicator	Interrupt indicator bit of channel 0. 0: No interrupt is generated. 1: An interrupt is generated.

VIn_STATUS

VIn_STATUS is the status register.

Offset Address	Register Name	Total Reset Value
0x004C+nx0x1000	VIn_STATUS	0x0000_0020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												act_height												vi_busy	field2	snooze	proc_err	bus_err	frame_loss	buf_ovf	image_done
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Access	Name	Description
[31:20]	RO	reserved	Reserved.
[19:8]	RO	act_height	Number of lines of detected valid pixel data in a field.
[7]	RO	vi_busy	Current working status of the VIU. 0: idle 1: busy.
[6]	RO	field2	Indicates whether even fields are received currently. 0: odd field 1: even field
[5]	RO	snooze	Indicates whether the VIU is in the snooze state. 0: non-snooze

			1: snooze
[4]	RO	proc_err	Protection bit error status 0: correct 1: incorrect
[3]	RO	bus_err	Bus error status. 0: correct 1: incorrect
[2]	RO	frame_loss	Indicates whether the VIU loses a field of data. 0: not lost 1: lost
[1]	RO	buf_ovf	VIU internal buffer overflow. 0: not overflow 1: overflow
[0]	RO	image_done	Indicates whether the receiving of current field of data is complete. 0: not complete 1: complete

VIn_LUM_ADDER

VIn_LUM_ADDER is the luminance adder register that is used to collect statistics on the luminance of all the active images instead of the luminance of the captured image.

Offset Address	Register Name	Total Reset Value
0x0050+nx0x1000	VIn_LUM_ADDER	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	vi_lum_adder	Accumulated luminance value.

NOTE

If the image block mask function is enabled, the luminance of the masked image block is insensitive to this register. In other words, when this register collects statistics on the luminance of the masked area, the statistics on the luminance of the masked image instead of the masked block is collected.

VIn_LUM_STRH

VIn_LUM_STRH is the luminance stretch register that is used to adjust the luminance of input images. The formula is as follows:

$$luma\_out = (sign(luma\_in - m0) * (k * | luma\_in - m0 | +64) >> 7) + m0$$

Where, m0 indicates the average luminance of the previous frame or field. It is calculated by dividing the accumulated luminance value (register VIn_LUM_ADDER) by the number of pixels of input images. The number of pixels refers to the number of received pixels instead of the number of pixels of the captured images.

The letter *k* indicates the stretch factor and ranges from 64 to 196. The value 128 indicates that the luminance difference between adjacent pixels keeps unchanged. If the value of k ranges from 64 to 128, the luminance difference between adjacent pixels is decreased; if the value of k ranges from 128 to 196, the luminance difference between adjacent pixels is increased.

It is recommended to set m1 and m0 to a same value.

Offset Address	Register Name	Total Reset Value
0x0054+nx0x1000	VIn_LUM_STRH	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved		m1										reserved		m0										k							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:30]	RO	reserved	Reserved.
[29:20]	RW	m1	Average luminance value 1.
[19:18]	RO	reserved	Reserved.
[17:8]	RW	m0	Average luminance value 0.
[7:0]	RW	k	Luminance stretch factor.

VIn_LUM_DIFF_ADDER

VIn_LUM_DIFF_ADDER is the luminance difference adder register. Its value is the sum of the absolute values of the differences between the luminance value of each input image and the average value m1 of VIn_LUM_STRH. VIn_LUM_DIFF_ADDER collects statistics on the luminance of all the active images but not only the luminance of the captured image. In frame storage mode, this register collects statistics by frames; in field storage mode, this register collects statistics by fields. The luminance of an input image refers to the luminance before down scaling. You can determine whether to enable the luminance stretch function based on this register.

- If the value of this register is large, it indicates that the luminance difference between the adjacent pixels of the previous frame or field is great. In this case, if you want to decrease the luminance difference, set the factor k of VIn_LUM_STRH to a value less than 128.
- If the value of this register is small, it indicates that the luminance difference is small. In this case, if you want to increase it, you can set the factor k of VIn_LUM_STRH to a value greater than 128.

Offset Address	Register Name	Total Reset Value
0x0058+nx0x1000	VIn_LUM_DIFF_ADDER	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	lum_diff_adder	Accumulated luminance difference value. In 720p HD mode, the accumulated luminance difference value of channel 0 is equal to

			that of input images. The accumulated value of channel 4 is invalid. In addition, the value of this register can be written only when the next frame or field occurs.

VIn_BLOCK0_START

VIn_BLOCK0_START is the start position register of image block 0.

Offset Address	Register Name	Total Reset Value
0x005C+nx0x1000	VIn_BLOCK0_START	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block0_starty												block0_startx											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	block0_starty	Number of the line from which image block 0 starts to be filled.
[11:0]	RW	block0_startx	Number of the pixel from which image block 0 starts to be filled.

VIn_BLOCK1_START

VIn_BLOCK1_START is the start position register of image block 1.

Offset Address	Register Name	Total Reset Value
0x0060+nx0x1000	VIn_BLOCK1 _START	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block1_starty												block1_startx											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	block1_starty	Number of the line from which image block 1 starts to be filled.
[11:0]	RW	block1_startx	Number of the pixel from which image block 1 starts to be filled.

VIn_BLOCK2_START

VIn_BLOCK2_START is the start position register of image block 2.

Offset Address	Register Name	Total Reset Value
0x0064+nx0x1000	VIn_BLOCK2 _START	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block2_starty												block2_startx											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	name	description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	block2_starty	Number of the line from which image block 2 starts to be filled.
[11:0]	RW	block2_startx	Number of the pixel from which image block 2 starts to be filled.

VIn_BLOCK3_START

VIn_BLOCK3_START is the start position register of image block 3.

Offset Address	Register Name	Total Reset Value
0x0068+nx0x1000	VIn_BLOCK3_START	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block3_starty												block3_startx											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	name	description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	block3_starty	Number of the line from which image block 3 starts to be filled.
[11:0]	RW	block3_startx	Number of the pixel from which image block 3 starts to be filled.

VIn_BLOCK0_SIZE

VIn_BLOCK0_SIZE is the size register of image block 0.

Offset Address	Register Name	Total Reset Value
0x006C+nx0x1000	VIn_BLOCK0_SIZE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block0_height												block0_width											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	description

[31:24]	RO	reserved	Reserved.
[23:12]	RW	block0_height	Height of image block 0 to be filled (in lines).
[11:0]	RW	block0_width	Width of image block 0 to be filled (in pixels).

VIn_BLOCK1_SIZE

VIn_BLOCK1_SIZE is the size register of image block 1.

Offset Address	Register Name	Total Reset Value
0x0070+nx0x1000	VIn_BLOCK1_SIZE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block1_height																block1_width							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	block1_height	Height of image block 1 to be filled (in lines).
[11:0]	RW	block1_width	Width of image block 1 to be filled (in pixels).

VIn_BLOCK2_SIZE

VIn_BLOCK2_SIZE is the size register of image block 2.

Offset Address	Register Name	Total Reset Value
0x0074+nx0x1000	VIn_BLOCK2_SIZE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block2_height																block2_width							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	block2_height	Height of image block 2 to be filled (in lines).
[11:0]	RW	block2_width	Width of image block 2 to be filled (in pixels).

VIn_BLOCK3_SIZE

VIn_BLOCK3_SIZE is the size register of image block 3.

Offset Address	Register Name	Total Reset Value
0x0078+nx0x1000	VIn_BLOCK3_SIZE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block3_height																block3_width							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	block3_height	Height of image block 3 to be filled (in lines).
[11:0]	RW	block3_width	Width of image block 3 to be filled (in pixels).

VIn_BLOCK0_COLOR

VIn_BLOCK0_COLOR is the filling color register of image block 0.

Note the following points when filling colors:

- The colors for filling images must be in the format of YCbCr.
- If multiple mask blocks are configured for the VIU and the blocks are overlapped with each other, the colors of the overlapped part are blended based on the priorities 0–3 in sequence. That is, blending block 0 takes the highest priority, whereas blending block 3 takes the lowest priority.

Offset Address	Register Name	Total Reset Value
0x007C+nx0x1000	VIn_BLOCK0_COLOR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block0_y								block0_u								block0_v							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:16]	RW	block0_y	Y component of the filling color of image block 0.
[15:8]	RW	block0_u	Cb component of the filling color of image block 0.
[7:0]	RW	block0_v	Cr component of the filling color of image block 0.

VIn_BLOCK1_COLOR

VIn_BLOCK0_COLOR is the filling color register of image block 1.

Offset Address	Register Name	Total Reset Value

0x0080+nx0x1000	VIn_BLOCK1_COLOR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block1_y								block1_u								block1_v							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:16]	RW	block1_y	Y component of the filling color of image block 1.
[15:8]	RW	block1_u	Cb component of the filling color of image block 1.
[7:0]	RW	block1_v	Cr component of the filling color of image block 1.

VIn_BLOCK2_COLOR

VIn_BLOCK2_COLOR is the filling color register of image block 2.

Offset Address	Register Name	Total Reset Value
0x0084+nx0x1000	VIn_BLOCK2_COLOR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block2_y								block2_u								block2_v							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:16]	RW	block2_y	Y component of the filling color of image block 2.
[15:8]	RW	block2_u	Cb component of the filling color of image block 2.
[7:0]	RW	block2_v	Cr component of the filling color of image block 2.

VIn_BLOCK3_COLOR

VIn_BLOCK3_COLOR is the filling color register of image block 3.

Offset Address	Register Name	Total Reset Value
0x0088+nx0x1000	VIn_BLOCK3_COLOR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								block3_y								block3_u								block3_v							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:16]	RW	block3_y	Y component of the filling color of image block 3.
[15:8]	RW	block3_u	Cb component of the filling color of image block 3.
[7:0]	RW	block3_v	Cr component of the filling color of image block 3.

VIn_ANC0_START

VIn_ANC0_START is the start position register of blanking area data block 0.

Offset Address	Register Name	Total Reset Value
0x009C+nx0x1000	VI_ANC0_START	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						anc0_loc		anc0_vos												anc0_hos											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25:24]	RW	anc0_loc	Blanking area containing data. 00: odd front blanking area 01: odd back blanking area 10: even front blanking area 11: even back blanking area **Note**: In progressive mode, there is only the odd front blanking area. In DC mode, the blanking area data cannot be received.
[23:12]	RW	anc0_vos	Distance from the start of the blanking area to the line containing the blanking area data (in lines).
[11:0]	RW	anc0_hos	Horizontal offset from the start of the line valid data to the start of the blanking area data (in clock cycles).

VIn_ANC0_SIZE

VIn_ANC0_SIZE is the size register of blanking area data block 0.

Offset Address	Register Name	Total Reset Value
0x00A0+nx0x1000	VIn_ANC0_SIZE	0x0000_0020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved																				anc0_size												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Access	Name	Description
[31:12]	RO	Reserved	Reserved.
[11:0]	RW	anc0_size	Size of the blanking area data block (in clock).

VIn_ANC1_START

VIn_ANC1_START is the start position register of blanking area data block 1.

Offset Address	Register Name	Total Reset Value
0x00A4+nx0x1000	VIn_ANC1_START	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						anc1_loc		anc1_vos												anc1_hos											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25:24]	RW	anc1_loc	Blanking area containing data. 00: odd front blanking area 01: odd back blanking area 10: even front blanking area 11: even back blanking area **Note**: In progressive mode, there is only the odd front blanking area. In DC mode, the blanking area data cannot be received.
[23:12]	RW	anc1_vos	Distance from the start of the blanking area to the line containing the blanking area data (in lines).
[11:0]	RW	anc1_hos	Horizontal offset from the start of the line valid data to the start of the blanking area data (in clock cycles).

VIn_ANC1_SIZE

VIn_ANC1_SIZE is the size register of blanking area data block 1.

Offset Address	Register Name	Total Reset Value
0x00A8+nx0x1000	VIn_ANC1_SIZE	0x0000_0020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved																				anc1_size												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Access	Name	Description
[31:12]	RO	Reserved	Reserved.
[11:0]	RW	anc1_size	Size of the blanking area data block (in clock cycles).

VIn_ANC0_WORD

VIn_ANC0_WORD is register group 1 that is used to store the obtained 16-word blanking area data. Register group 1 contains eight 32-bit registers. Data is stored in the registers starting from VIn_ANC0_WORD0. The offset address of each register is as follows:

- 0x00AC: VIn_ANC0_WORD0
- 0x00B0: VIn_ANC0_WORD1
- 0x00B4: VIn_ANC0_WORD2
- 0x00B8: VIn_ANC0_WORD3
- 0x00BC: VIn_ANC0_WORD4
- 0x00C0: VIn_ANC0_WORD5
- 0x00C4: VIn_ANC0_WORD6
- 0x00C8: VIn_ANC0_WORD7

The following table describes the registers by taking VIn_ANC0_WORD0 as an example. The rest may be deduced by analogy.

Offset Address	Register Name	Total Reset Value
0x00AC+nx0x1000	VIn_ANC0_WORD	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vin_anc0_word																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	vin_anc0_word	Register for storing the first word blanking area data, which belongs to register group 1.

VIn_ANC1_WORD

VIn_ANC1_WORD is register group 2 that is used to store the obtained 16-word blanking area data. Register group 2 contains eight 32-bit registers. Data is stored in the registers starting from VIn_ANC1_WORD0. The offset address of each register is as follows:

- 0x00CC: VIn_ANC1_WORD0
- 0x00D0: VIn_ANC1_WORD1
- 0x00D4: VIn_ANC1_WORD2
- 0x00D8: VIn_ANC1_WORD3
- 0x00DC: VIn_ANC1_WORD4
- 0x00E0: VIn_ANC1_WORD5
- 0x00E4: VIn_ANC1_WORD6
- 0x00E8: VIn_ANC1_WORD7

The following table describes the registers by taking VIn_ANC1_WORD0 as an example. The rest may be deduced by analogy.

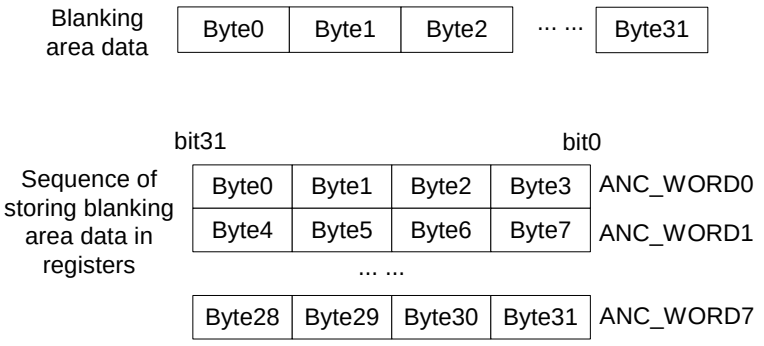
Offset Address	Register Name	Total Reset Value
0x00CC+nx0x1000	VIn_ANC1_WORD	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vin_anc1_word																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	vin_anc1_word	Register for storing the first word blanking area data, which belongs to register group 2.

Figure 6-31 shows the sequence of storing blanking area data in registers.

Figure 6-31 Sequence of storing blanking area data in registers



VI_P0_VSYNC1

When port 0 works in BT.601 mode, VI_P0_VSYNC1 is the vertical sync configuration register of field 1 and is valid for channel 0 only. When port 0 is connected to the 16-bit parallel interface and horizontal synchronization is performed in pulse mode, VI_P0_VSYNC1 is the vertical sync configuration register and is valid for channel 0 and channel 4.

Offset Address	Register Name	Total Reset Value
0x00EC	VI_P0_VSYNC1	0x0001_511F

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								act1_voff												act1_height											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	1	1	1	1

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	act1_voff	In BT.601 mode, act1_voff indicates the vertical distance from the start of field 1 to the active image; in 16-bit parallel interface mode, act1_voff indicates the vertical distance from the start of the frame to the active image. The distance is in the unit of line and its default value is 21. It is set to the value of subtracting 1 from the number of actual lines.
[11:0]	RW	act1_height	In BT.601 mode, act1_height indicates the height of active image of field 1; in 16-bit parallel interface mode, act1_height indicates the height of the active image. The height is in the unit of line and its default value is 287. It is set to the value of subtracting 1 from the number of actual lines.

NOTE

When the 16-bit parallel interface is used, the timing registers of the luminance channel and chrominance channel need to be configured and the configured values must be the same.

VI_P2_VSYNC1

When port 2 works in BT.601 mode, VI_P2_VSYNC1 is the vertical sync configuration register of field 1and is valid for channel 8 only. When port 2 is connected to the 16-bit parallel interface and horizontal synchronization is performed in pulse mode, VI_P2_VSYNC1 is the vertical sync configuration register and is valid for channel 8 and channel 12.

Offset Address	Register Name	Total Reset Value
0x80EC	VI_P0_VSYNC1	0x0001_511F

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								act1_voff												act1_height											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	1	1	1	1

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	act1_voff	In BT.601 mode, act1_voff indicates the vertical distance from the start of the active image of field 1; in 16-bit parallel interface mode, act1_voff indicates the vertical distance from the start of the frame to the active image. The distance is in the unit of line and its default value is 21. It is set to the value of subtracting 1 from the number of actual lines.
[11:0]	RW	act1_height	In BT.601 mode, act1_height indicates the height of the active image of field 1; in 16-bit parallel interface mode, act1_height indicates the height of the active image. The height is in the unit of line and its default value is 287. It is set to the value of subtracting 1 from the number of actual lines.

NOTE
When the 16-bit parallel interface is used, the timing registers of the luminance channel and chrominance channel need to be configured and the configured values must be the same.

VI_P0_VSYNC2

When port 0 works in BT.601 mode, VI_P0_VSYNC2 is the vertical sync configuration register of field 2 and is valid for channel 0 only.

Offset Address	Register Name	Total Reset Value
0x00F0	VI_P0_VSYNC2	0x0001_611F

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								act2_voff																act2_height							
Reset	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	act2_voff	act2_voff indicates the vertical distance from the start of field 2 to the active image. The distance is in the unit of line and its default value is 22. It is set to the value of subtracting 1 from the number of actual lines
[11:0]	RW	act2_height	act2_height indicates the height of the active image of field 2. The height is in the unit of line and its default value is 287. It is set to the value of subtracting 1 from the number of actual lines

NOTE
When the 16-bit parallel interface is used, the timing registers of the luminance channel and chrominance channel need to be configured and the configured values must be the same.

VI_P2_VSYNC2

When port 2 works in BT.601 mode, VI_P2_VSYNC2 is the vertical sync configuration register of field 2 and is valid for channel 8 only.

Offset Address	Register Name	Total Reset Value
0x80F0	VI_P0_VSYNC2	0x0001_611F

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								act2_voff																act2_height							
Reset	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	act2_voff	act2_voff indicates the vertical distance from the start of field 2 to the active image. The distance is in the unit of line and its default value is 22. It is set to the value of subtracting 1 from the number of actual lines
[11:0]	RW	act2_height	act2_height indicates the height of the active image of field 2. The height is in the unit of line and its default value is 287. It is set to the value of subtracting 1 from the number of actual lines

NOTE

When the 16-bit parallel interface is used, the timing registers of the luminance channel and chrominance channel need to be configured and the configured values must be the same.

VI_P0_HSYNC

VI_P0_HSYNC is the horizontal sync configuration register. When port 0 works in BT.601mode, this register is valid for channel 0 only; when port 0 is connected to the 16-bit parallel interface and horizontal synchronization is performed in pulse mode, this register is valid for channel 0 and channel 4.

Offset Address	Register Name	Total Reset Value
0x00F4	VI_P0_HSYNC	0x0004_159F

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

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.

[23:12]	RW	act_hoff	Distance from the end of the previous line to the active data area of the current line (in clock cycles). In BT.601 mode, the distance is 263 by default and can be set to the value of subtracting 1 from two times of the number of pixels. In 16-bit parallel interface mode, the distance can be set to the value of subtracting 1 from the number of pixels.
[11:0]	RW	act_width	Width of the active image (in clock cycles). In BT.601 mode, the width is 1439 (720 pixels) by default and can be set to the value of subtracting 1 from two times of the number of pixels. In 16-bit parallel interface mode, the distance can be set to the value of subtracting 1 from the number of pixels.

NOTE

When the 16-bit parallel interface is used, the timing registers of the luminance channel and chrominance channel need to be configured and the configured values must be the same.

VI_P2_HSYNC

VI_P2_HSYNC is the horizontal sync configuration register. When port 2 works in BT.601mode, this register is valid for channel 8 only; when port 2 is connected to the 16-bit parallel interface and horizontal synchronization is performed in pulse mode, this register is valid for channel 8 and channel 12.

Offset Address	Register Name	Total Reset Value
0x80F4	VI_P2_HSYNC	0x0004_159F

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

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:12]	RW	act_hoff	Distance from the end of the previous line to the active data area of the current line (in clock cycles). In BT.601 mode, the distance is 263 by default and can be set to the value of subtracting 1 from two times of the number of pixels. In 16-bit parallel interface mode, the distance can be set to the value of subtracting 1 from the number of pixels.

[11:0]	RW	act_width	Width of the active image (in clock cycles). In BT.601 mode, the width is 1439 (720 pixels) by default and can be set to the value of subtracting 1 from two times of the number of pixels. In 16-bit parallel interface mode, the distance can be set to the value of subtracting 1 from the number of pixels.

NOTE

When the 16-bit parallel interface is used, the timing registers of the luminance channel and chrominance channel need to be configured and the configured values must be the same.

VI_PRIO_CFG

VI_PRIO_CFG is the channel priority configuration register that is used to configure the priorities of the 16 internal channels of the VIU. Each channel supports two priorities.

Offset Address	Register Name	Total Reset Value
0x00F8	VI_PRIO_CFG	0x0004_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												outstanding_max				vi15_prio_ctrl	vi14_prio_ctrl	vi13_prio_ctrl	vi12_prio_ctrl	vi11_prio_ctrl	vi10_prio_ctrl	vi9_prio_ctrl	vi8_prio_ctrl	vi7_prio_ctrl	vi6_prio_ctrl	vi5_prio_ctrl	vi4_prio_ctrl	vi3_prio_ctrl	vi2_prio_ctrl	vi1_prio_ctrl	vi0_prio_ctrl
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:20]	RO	reserved	Reserved.
[19:16]	RW	outstanding_max	Maximum outstanding value of the AXI bus.
[15]	RW	vi15_prio_ctrl	Priority control bit of channel 15. 0: normal priority 1: high priority
[14]	RW	vi14_prio_ctrl	Priority control bit of channel 14. 0: normal priority 1: high priority
[13]	RW	vi13_prio_ctrl	Priority control bit of channel 13. 0: normal priority 1: high priority
[12]	RW	vi12_prio_ctrl	Priority control bit of channel 12. 0: normal priority 1: high priority
[11]	RW	vi11_prio_ctrl	Priority control bit of channel 11. 0: normal priority 1: high priority
[10]	RW	vi10_prio_ctrl	Priority control bit of channel 10. 0: normal priority 1: high priority
[9]	RW	vi9_prio_ctrl	Priority control bit of channel 9. 0: normal priority 1: high priority
[8]	RW	vi8_prio_ctrl	Priority control bit of channel 8. 0: normal priority 1: high priority
[7]	RW	vi7_prio_ctrl	Priority control bit of channel 7.

			0: normal priority 1: high priority
[6]	RW	vi6_prio_ctrl	Priority control bit of channel 6. 0: normal priority 1: high priority
[5]	RW	vi5_prio_ctrl	Priority control bit of channel 5. 0: normal priority 1: high priority
[4]	RW	vi4_prio_ctrl	Priority control bit of channel 4. 0: normal priority 1: high priority
[3]	RW	vi3_prio_ctrl	Priority control bit of channel 3. 0: normal priority 1: high priority
[2]	RW	vi2_prio_ctrl	Priority control bit of channel 2. 0: normal priority 1: high priority
[1]	RW	vi1_prio_ctrl	Priority control bit of channel 1. 0: normal priority 1: high priority
[0]	RW	vi0_prio_ctrl	Priority control bit of channel 0. 0: normal priority 1: high priority

VIn_LUM_COEF0

VIn_LUM_COEF0 is the luminance filtering coefficient register 0

The format of a filtering coefficient is as follows: The MSB is the sign bit and nine lower bits are the absolute values of the coefficient.

The sum of lum_coef0–lum_coef7 is 512.

During 1/2 down scaling, the recommended values of lum_coef0–lum_coef7 are –16, 0, 145, 254, 145, 0, –16, and 0 respectively.

Offset Address	Register Name	Total Reset Value
0x00FC+nx0x1000	VIn_LUM_COEF0	0x0000_0210

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						lum_coef1										reserved						lum_coef0									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25:16]	RW	lum_coef1	Luminance filtering coefficient 1. Its default value is 0.
[15:10]	RO	reserved	Reserved.
[9:0]	RW	lum_coef0	Luminance filtering coefficient 0. Its default value is –16.

VIn_LUM_COEF1

VIn_LUM_COEF1 is the luminance filtering coefficient register 1

The filtering coefficient is in the form of complement code and the MSB is sign bit.

Offset Address	Register Name	Total Reset Value
0x0100+nx0x1000	VIn_LUM_COEF1	0x00FE_0091

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						lum_coef3										reserved						lum_coef2									
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25:16]	RW	lum_coef3	Luminance filtering coefficient 3. Its default value is 254.
[15:10]	RO	reserved	Reserved.
[9:0]	RW	lum_coef2	Luminance filtering coefficient 2. Its default value is 145.

VIn_LUM_COEF2

VIn_LUM_COEF2 is the luminance filtering coefficient register 2

The filtering coefficient is in the form of complement code and the MSB is sign bit.

Offset Address	Register Name	Total Reset Value
0x0104+nx0x1000	VIn_LUM_COEF2	0x0000_0091

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						lum_coef5										reserved						lum_coef4									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25:16]	RW	lum_coef5	Luminance filtering coefficient 5. Its default value is 0.
[15:10]	RO	reserved	Reserved.
[9:0]	RW	lum_coef4	Luminance filtering coefficient 4. Its default value is 145.

VIn_LUM_COEF3

VIn_LUM_COEF3 is the luminance filtering coefficient register 3.

The filtering coefficient is in the form of complement code and the MSB is sign bit.

Offset Address	Register Name	Total Reset Value
0x0108+nx0x1000	VIn_LUM_COEF3	0x0000_0210

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						lum_coef7										reserved						lum_coef6									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25:16]	RW	lum_coef7	Luminance filtering coefficient 7. Its default value is 0.
[15:10]	RO	reserved	Reserved.
[9:0]	RW	lum_coef6	Luminance filtering coefficient 6. Its default value is –16.

VIn_CHROMA_COEF0

VIn_CHROMA_COEF0 is the chrominance filtering coefficient register 0.

The format of a filtering coefficient is as follows: The MSB is the sign bit and the nine lower bits are the absolute values of the coefficient.

The sum of chroma_coef0–chroma_coef3 is 512.

During 1/2 down scaling, the recommended values of chroma_coef0–chroma_coef3 are 148, 171, 148, and 45 respectively.

During chrominance resampling, the recommended values of chroma_coef0–chroma_coef3 are –32, 416, 160, and –32 respectively.

Offset Address	Register Name	Total Reset Value
0x010C+nx0x1000	VIn_CHROMA_COEF0	0x00AB_0094

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						chroma_coef1										reserved						chroma_coef0									
Reset	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25:16]	RW	chroma_coef1	Chrominance filtering coefficient 1. Its default value is 171.
[15:10]	RO	reserved	Reserved.
[9:0]	RW	chroma_coef0	Chrominance filtering coefficient 0. Its default value is 148.

VIn_CHROMA_COEF1

VIn_CHROMA_COEF1 is the chrominance filtering coefficient register 1.

The filtering coefficient is in the form of complement code and the MSB is sign bit.

Offset Address	Register Name	Total Reset Value
0x0110+nx0x1000	VIn_CHROMA_COEF1	0x002D_0094

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved						chroma_coef3										reserved						chroma_coef2									
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0

Bits	Access	Name	Description
[31:26]	RO	reserved	Reserved.
[25:16]	RW	chroma_coef3	Chrominance filtering coefficient 3. Its default value is 45.
[15:10]	RO	reserved	Reserved.

[9:0]	RW	chroma_coef2	Chrominance filtering coefficient 2. Its default value is 148.

6.2 VOU

6.2.1 Overview

The video output unit (VOU) reads the video data and graphic data from the specific positions of a memory, and then outputs the data through two display channels including the high-definition (HD) display channel and standard-definition (SD) display channel. It can process the graphic data of six layers at the same time including two video layers, three graphics layers, and one hardware cursor (HC) layer. In addition, the VOU supports various output interfaces through two display channels. Such output interfaces include BT.656 digital interface, composite video broadcast signal (CVBS) interface, and red-green-blue (RGB) analog interface.

6.2.2 Features

The VOU has the following features:

1. Supports two types of output interfaces

   Digital interfaces:

   - 8-bit ITU-R BT.656/YCbCr 4:2:2 standard output interface (PAL/NTSC standard @27 MHz)

   Analog interfaces:

   - M-NTSC or NTSC-J output interface
   - (B, D, G, H, I) PAL, (N) PAL, (Nc) PAL, or (M) PAL output interface
   - Video graphics array (VGA) output interface

- Processes the data of six layers at the same time

  HD video layer:

  - Supports three data read modes that can be configured through software: interlaced mode, progressive mode, and bottom field only mode
  - Supports two input pixel formats: semi-planar YCbCr4:2:2 and semi-planar YCbCr4:2:0
  - Supports the global alpha value
  - Supports horizontal and vertical scaling, that is, up to 2 x scaling down and 4 x scaling up
  - Supports 8-order horizontal luminance filtering and 4-order horizontal chrominance filtering in the formats of YCbCr4:2:2 and YCbCr4:2:0. For each filtering mode, there are 32 groups of configurable filtering coefficients.
  - Supports 4-order vertical luminance filtering and 4-order vertical chrominance filtering in the format of YCbCr4:2:2. For each filtering mode, there are 32 groups of configurable filtering coefficients.
  - Supports 4-order vertical luminance filtering and 2-order vertical chrominance filtering in the format of YCbCr4:2:0. For each filtering mode, there are 32 groups of configurable filtering coefficients.
  - Supports two de-interlace modes: 2-field median filtering and 4-field modes

- Supports the maximum input resolution of 1440 x 900 and the minimum input resolution of 32 x 32

SD video layer:

- Supports three data read modes that can be configured through software: interlaced mode, progressive mode, and bottom field only mode
- Supports two input pixel formats: semi-planar YCbCr4:2:2 and semi-planar YCbCr4:2:0
- Supports the global alpha value
- Supports the maximum input resolution of 720 x 576 and the minimum input resolution of 32 x 32

Graphics layer 0:

- Supports the 16-bit pixel formats ARGB1555, ARGB8888, and ARGB4444
- Supports the premultiplied pixel format
- Supports colorkey processing
- Supports global alpha and pixel alpha values
- Supporting BT.601 and BT.709 color space conversion

Graphics layer 1: Its features are the same as those of graphics layer 0.

Graphics layer 2: Its features are the same as those of graphics layer 0.

HC layer: Its features are the same as those of graphics layer 0.

- Supports blending functions
  - Supports two mixers: mixer 1 (for the HD display channel) and mixer 2 (for the SD display channel)
  - Mixer 1 supports the following blended layers: 24-bit background color layer, HD video layer, graphics layer 0, graphics layer 1, and HC layer. The background color layer has the lowest priority and the HC layer has the highest priority. The priorities of other layers are configurable.
  - Mixer 2 supports the following blended layers: 24-bit background color layer, SD video layer, graphics layer 1, graphics layer 2, and HC layer. The background color layer has the lowest priority and the HC layer has the highest priority. The priorities of other layers are configurable.
- Supports two display channels

  HD display channel
  - Supports the VGA analog interface, up to 1440 x 900@60Hz
  - Supports the conversion from YCbCr to RGB. The conversion coefficient is configurable.
  - Supports the adjustable contrast, luminance, and saturation
  - Supports dynamic gamma adjust

  SD display channel
  - Supports CVBS analog output (NTSC/PAL)
  - Supports the conversion from YCbCr to RGB. The conversion coefficient is configurable.
  - Supports the adjustable contrast, luminance, and saturation

6.2.3 Signal Description

Table 6-8 lists the signals of the VOU digital output interface.

Table 6-8 Signals of the VOU digital output interface

Signal	Direction	Description	Pin
clk_vo0out	O	BT.656 associated clock output. The normal phase output mode and reverse phase output mode are configurable.	VOCK
vo_vga0_hsync	O	Horizontal sync signal output of the HD channel	VGAHS
vo_vga0_vsync	O	Vertical sync signal output of the HD channel	VGAVS
vo_656[0]	O	BT.656 data output	VODAT0
vo_656[1]			VODAT1
vo_656[2]			VODAT2
vo_656[3]			VODAT3
vo_656[4]			VODAT4
vo_656[5]			VODAT5
vo_656[6]			VODAT6
vo_656[7]			VODAT7

Table 6-9 lists the signals of the VOU analog output interface.

Table 6-9 Signals of the VOU analog output interface

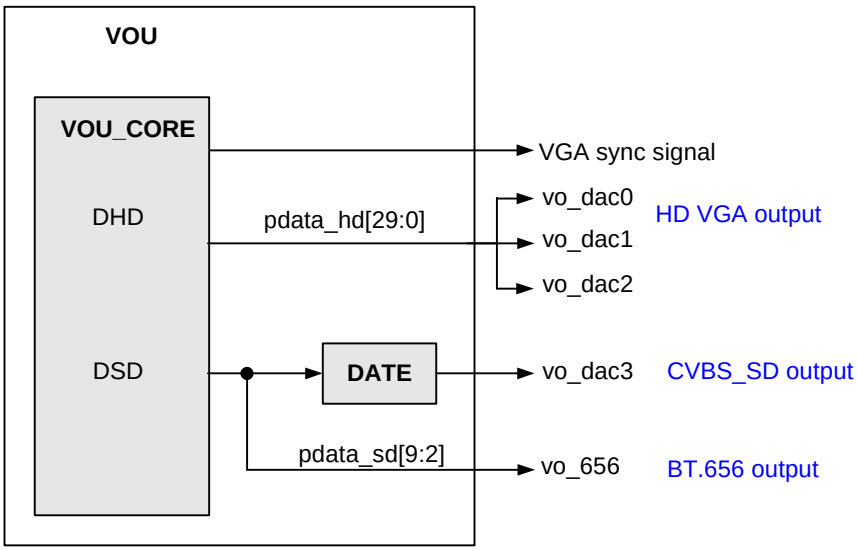
Signal	Direction	Description	Pin
vo_dac0	O	R component of the VGA, connected to DAC1 output	DACVGA1R
vo_dac1	O	G component of the VGA, connected to DAC1 output	DACVGA1G
vo_dac2	O	B component of the VGA, connected to DAC1 output	DACVGA1B
vo_dac3	O	CVBS, connected to DAC0 output	DACVGA0R

6.2.4 Function Description

VOU Output Interfaces

Figure 6-32 shows the multiplexing relationship between VOU interfaces. The interface multiplexing is controlled by configuring the VOMUX register.

Figure 6-32 Multiplexing relationship between VOU interfaces



Functions of Video Layers

1. De-interlace

The de-interlace algorithm is classified into two modes based on the requirements on the image quality:

- 2-field median filtering mode
- 4-field mode (motion estimation, time-domain filtering, spatio-temporal weight, and edge detection)

The 4-field mode involves full-bandwidth mode and half-width mode. The time-domain filtering is supported in full-bandwidth mode only.

The consumed system bandwidth varies according to the data amount to be processed. When a frame is output, 2-field data needs to be read in 2-field median filtering mode and 4-field data needs to be read in 4-field mode.

2. Scaling

The scaling unit of the VOU uses the 8-order, 4-order, and 2-order finite impulse response (FIR) filters with 32 phases. All filter coefficients are configurable.

- The horizontal luminance is scaled by using the 8-order FIR filter with 32 phases.
- The horizontal chrominance and vertical luminance are scaled by using the 4-order FIR filter with 32 phases.
- The vertical chrominance in the YCbCr4:2:2 format is scaled by using the 4-order FIR filter with 32 phases.
- The vertical chrominance in the YCbCr4:2:0 format is scaled by using the 2-order FIR filter with 32 phases.

Functions of the Graphics Layers

1. Alpha processing

The VOU has four graphics layers: graphics layer 0, graphics layer 1, graphics layer 2, and HC layer. The layers are blended according to their priorities. The blending alpha value has 129 levels and the sources of the alpha value is related to layers.

- The alpha value of video layers originates from only the global alpha value configured by registers.
- The alpha value of graphics layers can originate from the following two sources:
  - Pixel alpha value: blending attribute of a pixel.
  - Global alpha value: blending attribute of a layer.

When n (n ≥ 2) layers are blended, the preceding two attributes are considered at the same time. The alpha value of n blended layers is equal to the product of n alpha values. The global alpha value is obtained according to the configuration. When there is no pixel alpha value, its value is the maximum value 1 by default.

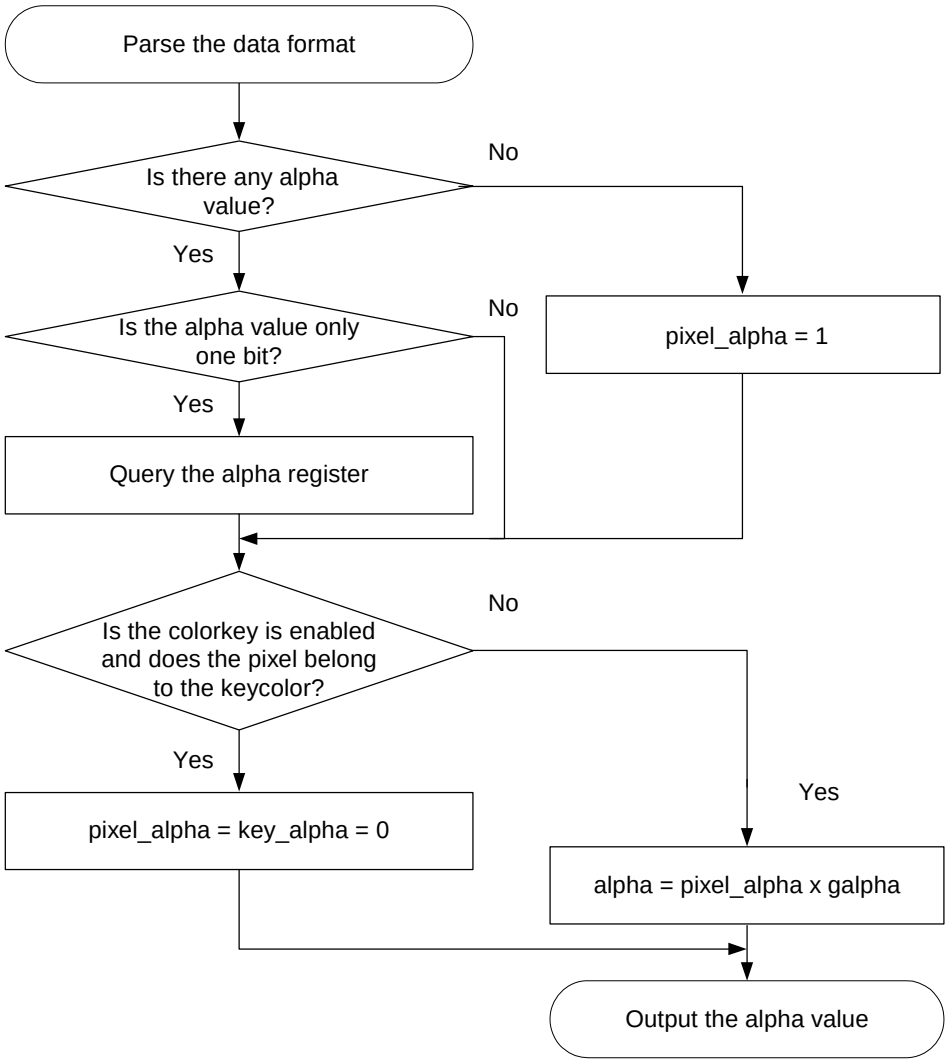
There is a special case for the pixel alpha value. In the RGB1555 format, the alpha value is only one bit. This bit is not the actual alpha value. It is the index of the alpha value that is used to select the actual alpha value from the alpha register. That is, the actual alpha value is selected based on the values of alpha0 and alpha1.

CAUTION

When a pixel is the colorkey, this pixel cannot be blended.

Figure 6-33 shows how to process the pixel alpha data of a graphics layer.

Figure 6-33 Processing the pixel alpha data of a graphics layer



NOTE

pixel_alpha: pixel alpha value

galpha: global alpha value

key_alpha: alpha value of the colorkey

2. Colorkey processing

At graphics layers, the color that is the same as the selected color is called the colorkey. The colorkey is configured by registers. When the colorkey is enabled, the hardware matches colors at corresponding layers. When the hardware finds the colorkey, the hardware replaces the alpha value of the color with 0. The part with the colorkey is processed as a transparent part.

Coordinates of Video Layers

Each video layer has three sets of coordinates, as shown in Figure 6-34. The coordinates are as follows:

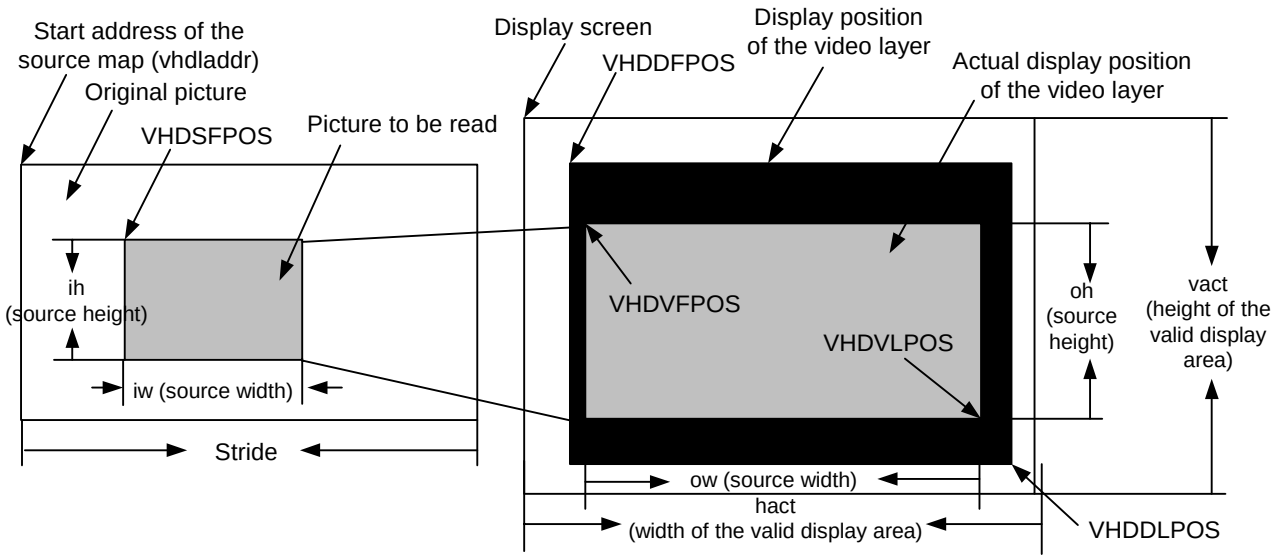
- Source start coordinates of the data to be read

- Start and end coordinates of the video to be displayed
- Start and end coordinates of the video displayed on the screen

CAUTION

- The source start coordinates must be set to (0, 0).
- The start and end coordinates of the video to be displayed and the start and end coordinates of the video displayed on the screen must be the same.

Figure 6-34 Three sets of coordinates of a video layer



Color Space Conversion

The VOU supports color space conversion from YCbCr to RGB or from RGB to YCbCr and color enhancement.

- Conversion from YCbCr to RGB

  The data format is YCbCr during blending. If the data is displayed in VGA, the format needs to be converted from YCbCr to RGB.

- Conversion from RGB to YCbCr

  If a RGB graphics layer is displayed on TV, its format needs to be converted into YCbCr 4:2:2.

The matrix of conversion from YCbCr to RGB is as follows:

$$\begin{bmatrix} R'255 \\ G'255 \\ B'255 \end{bmatrix} = \begin{bmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{bmatrix} \bullet \begin{bmatrix} Y'-in\_dc0 \\ Cb-in\_dc1 \\ Cr-in\_dc2 \end{bmatrix}$$

The matrix of conversion from RGB to YCbCr is as follows:

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} out\_dc0 \\ out\_dc1 \\ out\_dc2 \end{bmatrix} + \begin{bmatrix} coef00 & coef01 & coef02 \\ coef10 & coef11 & coef12 \\ coef20 & coef21 & coef22 \end{bmatrix} \bullet \begin{bmatrix} R'255 \\ G'255 \\ B'255 \end{bmatrix}$$

Interrupt

The VOU has 17 interrupt sources, one interrupt status register, and one interrupt mask register. When an interrupt source is masked, the interrupt status register still records the interrupt status, but does not report the interrupt.

The VOU supports the following interrupts:

- Register update interrupts of the HD video layer, SD video layer, graphics layer 0, graphics layer 1, graphics layer 2, and HC layer
- Write bus low-bandwidth interrupts of the HD video layer
- Bus error interrupt
- Low-bandwidth alarm interrupts of the HD display channel and SD display channel
- Vertical timing interrupts of the HD display channel and SD display channel

6.2.5 Operating Mode

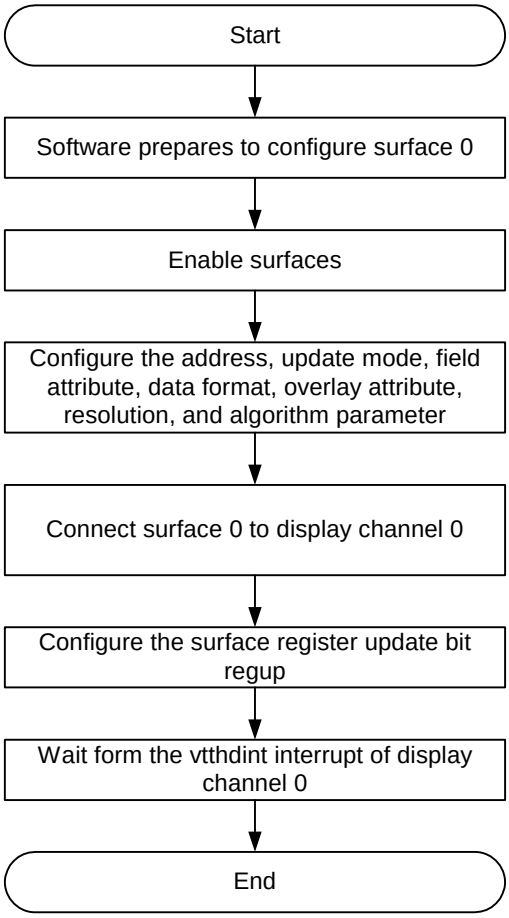
Updates of Surface Registers

Figure 6-35 shows the updates of surface registers (non-instant registers).

CAUTION

- Each surface must be connected to a display channel.
- After all the surface parameters are configured, the register update bit regup must be configured.

Figure 6-35 Process of configuring surface registers (recommended)



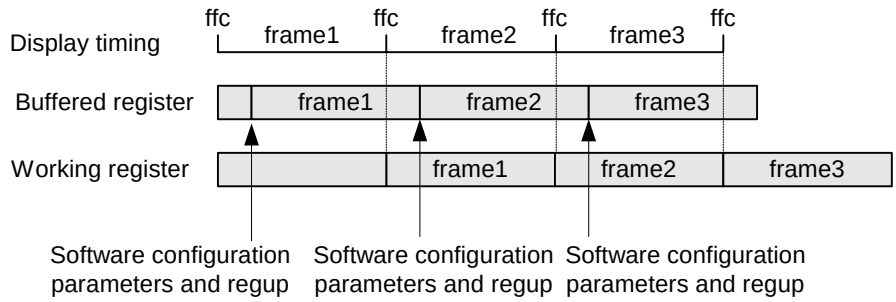
After surface 0 is connected to display channel 0, surface 0 updates its register based on the timing interrupt of display channel 0. Therefore, software can check whether update time of the register is reached based on the vtthdint interrupt of display channel 0.

Hardware has two types of surface registers:

- Working register: the register configuration of the current data channel
- buffered register

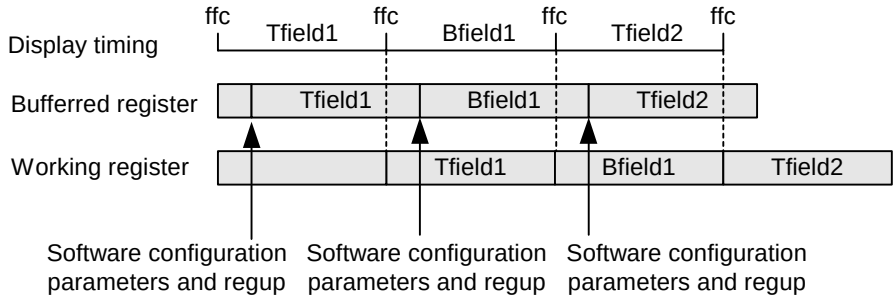
The register values configured by software are buffered in the buffered register. After the register update time is reached, the register values are imported to the working register. Figure 6-36 shows the frame update mode of surface registers.

Figure 6-36 Frame update mode of surface registers



The video layers support not only the frame update mode, but also the field update mode for interlaced output, as shown in Figure 6-37.
In interlaced output mode, you need to set the update modes of video layers to the field update mode by configuring the regup_rate bits of channel control registers. In Figure 6-37, ffc indicates the point of switching between frame and fields.

Figure 6-37 Field update mode of surface registers



Updata of Display Channel Registers

The display channel registers (instant registers) are classified into two types:

- Operation-related registers, including the registers that are related to CSC, CLIP, background color, and gamma. You can configure such registers at any time as required. To avoid the display exception of any pixel, you are recommended to configure them in the blanking area.
- Timing-control-related registers. To configure such registers, you must disable display channels before configuring the registers and then enable the channels after configuration.

Configuration Mode of Blending Registers

Six surfaces are allowed to connect to mixer 1 and mixer 2. A surface cannot drive two mixers at the same time. The mixer driven by a surface is configurable, but the mixer cannot be changed in real time. Before changing the mixer driven by a surface, you must disable all the related VO interface and surfaces. The configuration process is as follows:

Step 1 Disable related VO interfaces.

Step 2 Configure CBCFG bit[29:24] to query the mapping between each surface and mixer.

Step 3 Configure the registers of each surface.

Step 4 Configure the priority of each surface.

Step 5 Enable all the VO interface.

----End

You can configure the priorities of the surfaces that drive mixer 1 and mixer 2 by configuring CBCFG bit[14:0]. It is recommended to set a lower priority for the surface of mixer 1 and set a higher priority for the surface of mixer 2, because a surface cannot drive two mixers at the same time.

For example, the surfaces for driving mixer 1 include VDC_HD, GDC_G0, GDC_G2 that have the priorities from low to high. The surfaces for driving mixer 2 include VDC_SD and GDC_G1 that also have the priorities from low to high. You can configure the priorities as follows:

- VDC_HD, GDC_G0, and GDC_G2 drive mixer 1.

```
sur_attr0 = 0; sur_attr2 = 0; sur_attr4 = 0;
```

- VDC_SD and GDC_G1 drive mixer 2.

```
sur_attr1 = 1; sur_attr3 = 1;
```

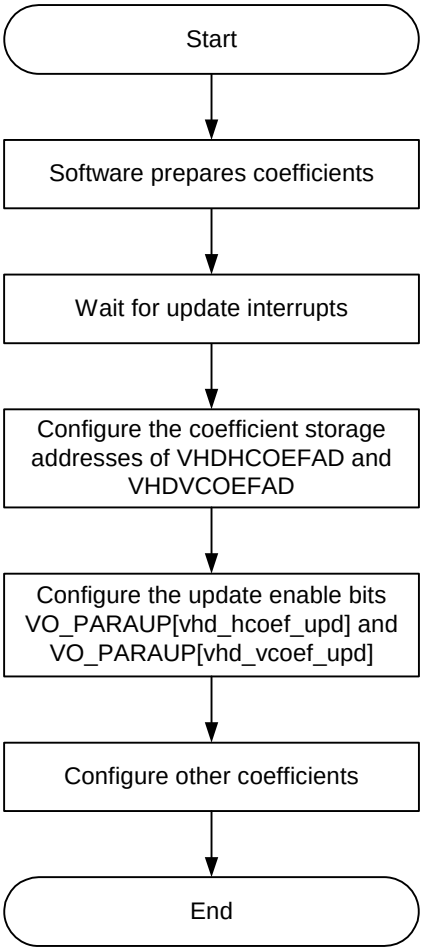
- Configure priorities.

```
mixer_prio0 = 1; mixer_prio1 = 3; mixer_prio2 =5; mixer_prio3 =
2;mixer_prio4 =4;
```

Storage and Updates of On-Chip Coefficients

The data amount of on-chip coefficients is large. To avoid additional CPU usage, the software does not configure the coefficients through the advanced peripheral bus (APB) slave. Before the hardware automatically updates the coefficients by reading the AXI master, the software needs to configure the coefficient update enable bits and coefficient storage addresses. After updates, the software adds the addresses and update commands to the coefficient update queue. Then AXI master updates the addresses and commands. Therefore, the software can configure multiple coefficient update commands continuously. Figure 6-38 shows how to update the on-chip coefficients. The on-chip coefficients can be read through the APB slave.

Figure 6-38 Updating the on-chip coefficients



Disabling of the Clock

Before disabling the VOU bus clock, you must disable all the interfaces. Otherwise, the interrupt status cannot be cleared after the bus is disabled.

6.2.6 Register Summary

Table 6-10 shows the VOU registers.

Table 6-10 Summary of VOU registers (base address: 0x2013_0000)

Offset Address	Register	Description	Page
0x0000	VO_CTRL	VO control register	6-98
0x0004	VO_INTSTA	VO interrupt status register	6-98
0x0008	VO_INTMSK	VO interrupt mask register	6-100
0x000C	VO_VERSION1	VO version register 1	6-102
0x0010	VO_VERSION2	VO version register 2	6-102

Offset Address	Register	Description	Page
0x002C	VO_PARAUP	Scaling and gamma coefficient update enable register	6-102
0x0030	VHDHCOEFAD	Horizontal luminance and chrominance filtering coefficient address register of the VHD channel	6-103
0x0034	VHDVCOEFAD	Vertical luminance and horizontal chrominance filtering coefficient address register of the VHD channel	6-104
0x0040	DHDACCAD	ACC coefficient lookup table address register of the DHD channel	6-104
0x0100	VHDCTRL	VHD channel control register	6-105
0x0104	VHDUPD	VHD channel update enable register	6-106
0x0108	VHDLADDR	Address register of the previous de-interlace frame of the VHD channel	6-106
0x010C	VHDLCADDR	Chrominance address register of the previous de-interlace frame of the VHD channel	6-107
0x0110	VHDCADDR	Address register of the current de-interlace frame of the VHD channel	6-107
0x0114	VHDCCADDR	Chrominance address register of the current de-interlace frame of the VHD channel	6-107
0x0118	VHDNADDR	Address register of the next de-interlace frame of the VHD channel	6-108
0x011C	VHDNCADDR	Chrominance address register of the next de-interlace frame of the VHD channel	6-108
0x0120	VHDSTRIDE	Line stride register of the VHD channel	6-109
0x0124	VHDCBMPARA	Blending parameter register of the VHD channel (non-instant register)	6-109
0x0128	VHDORESO	Output resolution register of the VHD channel (non-instant register)	6-109

Offset Address	Register	Description	Page
0x012C	VHDIRESO	Input resolution register of the VHD channel (non-instant register)	6-110
0x0130	VHDSFPOS	Source bitmap start position register of the VHD channel (non-instant register)	6-111
0x0134	VHDDFPOS	Display window start position register of the VHD channel (non-instant register, in pixels)	6-111
0x0138	VHDDLPOS	Display window end position register of the VHD channel (non-instant register, in pixels)	6-112
0x013C	VHDVFPOS	Video start position register in the display window of the VHD channel (non-instant register, in pixels)	6-112
0x0140	VHDVLPOS	Video end position register in the display window of the VHD channel (non-instant register, in pixels)	6-113
0x0144	VHDBK	Video layer background color register of the VHD channel	6-113
0x0150	VHDLMSP	Luminance scaling parameter configuration register of the VHD channel	6-114
0x0154	VHDCHMSP	Chrominance scaling parameter configuration register of the VHD channel	6-115
0x0158	VHDLMHSP	Luminance horizontal scaling parameter configuration register of the VHD channel (non-instant register)	6-116
0x015C	VHDLMVSP	Luminance vertical scaling parameter configuration register of the VHD channel (non-instant register)	6-116
0x0160	VHDCHMHSP	Chrominance horizontal scaling parameter configuration register of the VHD channel	6-117
0x0164	VHDCHMVSP	Chrominance vertical scaling parameter configuration register of the VHD channel	6-117

Offset Address	Register	Description	Page
0x0170	VHDDIECTRL	De-interlace operation control register of the VHD channel (non-instant register)	6-118
0x0174	VHDDIETHD	De-interlace operation threshold register of the VHD channel (non-instant register)	6-120
0x0178	VHDDIEADDR	De-interlace history buffer address register of the VHD channel (non-instant register)	6-120
0x0180+n1×0x4	VHDDIETSMIX	De-interlace spatio-temporal weight coefficient register of the VHD channel (instant register)	6-121
0x0190+n2×0x4	VHDDIETFLT	De-interlace time-domain filtering coefficient register of the VHD channel (instant register)	6-121
0x01A0+n3×0x4	VHDDIEVFLT	De-interlace vertical filtering weight coefficient register of the VHD channel (instant register)	6-122
0x01F0	VHDSTATUS	Video layer status register of the VHD channel	6-123
0x0300	VSDCTRL	VSD channel control register	6-123
0x0304	VSDUPD	VSD channel update enable register	6-125
0x0310	VSDADDR	Address register of the current frame of the VSD channel	6-125
0x0314	VSDCADDR	Chrominance address register of the current frame of the VSD channel	6-125
0x0320	VSDSTRIDE	Line stride register of the VSD channel	6-126
0x0324	VSDCBMPARA	Blending parameter register of the VSD channel (non-instant register)	6-126
0x0328	VSDORESO	Output resolution register of the VSD channel (non-instant register)	6-127
0x032C	VSDIRESO	Input resolution register of the VSD channel (non-instant register)	6-128

Offset Address	Register	Description	Page
0x0330	VSDSFPOS	Source bitmap start position register of the VSD channel (non-instant register)	6-128
0x0334	VSDDFPOS	Display window start position register of the VSD channel (non-instant register, in pixels)	6-129
0x0338	VSDDLPOS	Display window end position register of the VSD channel (non-instant register, in pixels)	6-129
0x033C	VSDVFPOS	Video start position register in the display window of the VSD channel (non-instant register, in pixels)	6-130
0x0340	VSDVLPOS	Video end position register in the display window of the VSD channel (non-instant register, in pixels)	6-130
0x0344	VSDBK	Video layer background color register of the VSD channel	6-131
0x0400	G0CTRL	Graphics layer 0 (G0) channel control register (non-instant register)	6-131
0x0404	G0UPD	G0 channel update enable register	6-132
0x0408	G0ADDR	Frame address register of the G0 channel	6-133
0x040C	G0STRIDE	Frame stride register of the G0 channel	6-133
0x0410	G0CBMPARA	Blending parameter register of the G0 channel (non-instant register)	6-134
0x0414	G0CKEYMAX	Maximum colorkey value register of the G0 channel (non-instant register)	6-135
0x0418	G0CKEYMIN	Minimum colorkey value register of the G0 channel (non-instant register)	6-135
0x041C	G0IRESO	Input resolution register of the G0 channel (non-instant register)	6-136
0x0420	G0ORESO	Output resolution register of the G0 channel (non-instant register)	6-137

Offset Address	Register	Description	Page
0x0424	G0SFPOS	Source bitmap start position register of the G0 channel (non-instant register)	6-137
0x0428	G0DFPOS	Display window start position register of the G0 channel (non-instant register, in pixels)	6-138
0x042C	G0DLPOS	Display window end position register of the G0 channel (non-instant register, in pixels)	6-138
0x0500	G1CTRL	Graphics layer 1 (G1) channel control register (non-instant register)	6-139
0x0504	G1UPD	G1 channel update enable register	6-139
0x0508	G1ADDR	Frame address register of the G1 channel	6-140
0x050C	G1STRIDE	Frame stride register of the G1 channel	6-140
0x0510	G1CBMPARA	Blending parameter register of the G1 channel (non-instant register)	6-141
0x0514	G1CKEYMAX	Maximum colorkey value register of the G1 channel (non-instant register)	6-142
0x0518	G1CKEYMIN	Minimum colorkey value register of the G1 channel (non-instant register)	6-142
0x051C	G1IRESO	Input resolution register of the G1 channel (non-instant register)	6-143
0x0520	G1ORESO	Output resolution register of the G1 channel (non-instant register)	6-144
0x0524	G1SFPOS	Source bitmap start position register of the G1 channel (non-instant register)	6-144
0x0528	G1DFPOS	Display window start position register of the G1 channel (non-instant register, in pixels)	6-145
0x052C	G1DLPOS	Display window end position register of the G1 channel (non-instant register, in pixels)	6-145

Offset Address	Register	Description	Page
0x0600	G2CTRL	Graphics layer 2 (G2) channel control register (non-instant register)	6-146
0x0604	G2UPD	G2 channel update enable register	6-146
0x0608	G2ADDR	Frame address register of the G2 channel	6-147
0x060C	G2STRIDE	Frame stride register of the G2 channel	6-147
0x0610	G2CBMPARA	Blending parameter register of the G2 channel (non-instant register)	6-148
0x0614	G2CKEYMAX	Maximum colorkey value register of the G2 channel (non-instant register)	6-149
0x0618	G2CKEYMIN	Minimum colorkey value register of the G2 channel (non-instant register)	6-149
0x061C	G2IRESO	Input resolution register of the G2 channel (non-instant register)	6-150
0x0620	G2ORESO	Output resolution register of the G2 channel (non-instant register)	6-151
0x0624	G2SFPOS	Source bitmap start position register of the G2 channel (non-instant register)	6-151
0x0628	G2DFPOS	Display window start position register of the G2 channel (non-instant register, in pixels)	6-152
0x062C	G2DLPOS	Display window end position register of the G2 channel (non-instant register, in pixels)	6-152
0x0800	HCCTRL	HC channel control register (non-instant register)	6-153
0x0804	HCUPD	HC channel update enable register	6-153
0x0808	HCADDR	Frame address register of the HC channel	6-154
0x080C	HCSTRIDE	Frame stride register of the HC channel	6-154

Offset Address	Register	Description	Page
0x0810	HCCBMPARA	Blending parameter register of the HC channel (non-instant register)	6-155
0x0814	HCCKEYMAX	Maximum colorkey value register of the HC channel (non-instant register)	6-156
0x0818	HCCKEYMIN	Minimum colorkey value register of the HC channel (non-instant register)	6-156
0x081C	HCIRESO	Input resolution register of the HC channel (non-instant register)	6-157
0x0820	HCORESO	Output resolution register of the HC channel (non-instant register)	6-158
0x0824	HCSFPOS	Source bitmap start position register of the HC channel (non-instant register)	6-158
0x0828	HCFPOS	Display window start position register of the HC channel (non-instant register, in pixels)	6-159
0x082C	HCDLPOS	Display window end position register of the HC channel (non-instant register, in pixels)	6-159
0x0B00	CBMBKG1	Blending background color register of mixer 1 (instant register)	6-160
0x0B04	CBMBKG2	Blending background color register of mixer 2 (instant register)	6-160
0x0B0C	CBCFG	Color bar configuration register (instant register)	6-160
0x0C00	DHDCTRL	HD display channel control register (instant register)	6-163
0x0C04	DHDVSYNC	Vertical sync timing register of the HD display channel (instant register)	6-165
0x0C08	DHDHSYNC1	Horizontal sync timing register 1 of the HD display channel (instant register)	6-165
0x0C0C	DHDHSYNC2	Horizontal sync timing register 2 of the HD display channel (instant register)	6-166

Offset Address	Register	Description	Page
0x0C10	DHDVPLUS	Vertical sync plus timing register of the HD display channel (instant register)	6-166
0x0C14	DHDPWR	Sync pulse width register of the HD display channel (instant register)	6-167
0x0C18	DHDFIFOTHD	Async FIFO threshold configuration register of the HD display channel (instant register)	6-167
0x0C1C	DHDVTTHD	Vertical timing threshold register of the HD display channel (instant register)	6-167
0x0C20	DHDCSCIDC	Input DC component for color space conversion register of the HD display channel (instant register)	6-168
0x0C24	DHDCSCODC	Output DC component for color space conversion register of the HD display channel (instant register)	6-169
0x0C28	DHDCSCP0	Color space conversion parameter 0 register of the HD display channel (instant register)	6-169
0x0C2C	DHDCSCP1	Color space conversion parameter 1 register of the HD display channel (instant register)	6-170
0x0C30	DHDCSCP2	Color space conversion parameter 2 register of the HD display channel (instant register)	6-170
0x0C34	DHDCSCP3	Color space conversion parameter 3 register of the HD display channel (instant register)	6-171
0x0C38	DHDCSCP4	Color space conversion parameter 4 register of the HD display channel (instant register)	6-171
0x0C3C	DHDCLIPL	Minimum threshold clip bit processing register of the HD display channel (instant register)	6-172
0x0C40	DHDCLIPH	Maximum threshold clip bit processing register of the HD display channel (instant register)	6-172

Offset Address	Register	Description	Page
0x0C44	DHDGMMTHD1	Gamma operation threshold 1 register of the HD display channel (instant register)	6-173
0x0C48	DHDGMMTHD2	Gamma operation threshold 2 register of the HD display channel (instant register)	6-173
0x0C50+t1×0x4	DHDGMMLOWt	Low-luminance gamma lookup table register of the HD display channel (instant register)	6-173
0x0C60+t2×0x4	DHDGMMMEDt	Medium-luminance gamma lookup table register of the HD display channel (instant register)	6-174
0x0C70+t3×0x4	DHDGMMHIGHt	High-luminance gamma lookup table register of the HD display channel (instant register)	6-174
0x0C80+t4×0x4	DHDGMMMLt	Middle_low-luminance gamma lookup table register of the HD display channel (instant register)	6-175
0x0CA0+t5×0x4	DHDGMMMHt	Middle_high-luminance gamma lookup table register of the HD display channel (instant register)	6-175
0x0CB0	DHDGMM3LOW	Low-threshold luminance statistics register of three gamma areas of the HD display channel	6-176
0x0CB4	DHDGMM3MED	Medium-threshold luminance statistics register of three gamma areas of the HD display channel	6-176
0x0CB8	DHDGMM3HIGH	High-threshold luminance statistics register of three gamma areas of the HD display channel	6-177
0x0CC0	DHDGMM8MLOW	Low-threshold luminance statistics register after the middle gamma area of the HD display channel is divided into eight segments	6-177
0x0CC4	DHDGMM8MHIGH	High-threshold luminance statistics register after the middle gamma area of the HD display channel is divided into eight segments	6-178
0x0CF0	DHDSTATE	HD display channel status register	6-178

Offset Address	Register	Description	Page
0x0E00	DSDCTRL	SD display channel control register (instant register)	6-179
0x0E04	DSDVSYNC	Vertical sync timing register of the SD display channel (instant register)	6-180
0x0E08	DSDHSYNC1	Horizontal sync timing register 1 of the SD display channel (instant register)	6-181
0x0E0C	DSDHSYNC2	Horizontal sync timing register 2 of the SD display channel (instant register)	6-181
0x0E10	DSDVPLUS	Vertical sync plus timing register of the SD display channel (instant register)	6-182
0x0E14	DSDPWR	Sync pulse width register of the SD display channel (instant register)	6-182
0x0E18	DSDFIFOTHD	Async FIFO threshold configuration register of the SD display channel (instant register)	6-183
0x0E1C	DSDVTTHD	Vertical timing threshold register of the SD display channel (instant register)	6-183
0x0E20	DSDCSCIDC	Input DC component for color space conversion register of the SD display channel (instant register)	6-184
0x0E24	DSDCSCODC	Output DC component for color space conversion register of the SD display channel (instant register)	6-185
0x0E28	DSDCSCP0	Color space conversion parameter 0 register of the SD display channel (instant register)	6-185
0x0E2C	DSDCSCP1	Color space conversion parameter 1 register of the SD display channel (instant register)	6-186
0x0E30	DSDCSCP2	Color space conversion parameter 2 register of the SD display channel (instant register)	6-186
0x0E34	DSDCSCP3	Color space conversion parameter 3 register of the SD display channel (instant register)	6-187

Offset Address	Register	Description	Page
0x0E38	DSDCSCP4	Color space conversion parameter 4 register of the SD display channel (instant register)	6-187
0x0E3C	DSDCLIPL	Minimum threshold clip bit processing register of the SD display channel (instant register)	6-188
0x0E40	DSDCLIPH	Maximum threshold clip bit processing register of the SD display channel (instant register)	6-188
0x0EF0	DSDSTATE	SD display channel status register	6-189
0x1000–0x111C	VHDHLCOEF	Luminance horizontal scaling and filtering coefficient register of the VHD channel	6-189
0x1200–0x128C	VHDHCCOEF	Chrominance horizontal scaling and filtering coefficient register of the VHD channel	6-190
0x1300–0x138C	VHDVLCOEF	Luminance vertical scaling and filtering coefficient register of the VHD channel	6-191
0x1400–0x148C	VHDVCCOEF	Chrominance vertical scaling and filtering coefficient register of the VHD channel	6-191
0x2300–0x237C	VHDMIMGSPOSp	Start position and valid flag register of subimage p in the de-interlace partition of the VHD channel	6-192
0x2380–0x23FC	VHDMIMGFPOSp	End position register of subimage p in the de-interlace partition of the VHD channel	6-193
0x2600	DHDVBISPOS1	Position and start coordinate register of VBI information 1 of the DHD channel	6-193
0x2604	DHDVBISPOS2	Position and start coordinate register of VBI information 2 of the DHD channel	6-194
0x2610–0x262C	DHDVBIF1n	Content register of VBI information 1 of the DHD channel	6-195
0x2630–0x264C	DHDVBIF2n	Content register of VBI information 2 of the DHD channel	6-196

Offset Address	Register	Description	Page
0x2700	DSDVBISPOS1	Position and start coordinate register of VBI information 1 of the DSD channel	6-196
0x2704	DSDVBISPOS2	Position and start coordinate register of VBI information 2 of the DSD channel	6-197
0x2710–0x272C	DSDVBIF1n	Content register of VBI information 1 of the DSD channel	6-198
0x2730–0x274C	DSDVBIF2n	Content register of VBI information 2 of the DSD channel	6-199

Table 6-11 lists the value range and meaning of the variables in the offset addresses of VOU registers.

Table 6-11 Variables in the offset addresses of VOU registers

Variable	Value Range	Description
n1	0–3	Indicates the group ID of de-interlace spatio-temporal weight coefficients
n2	0–3	Indicates the group ID of de-interlace time-domain filtering coefficients
n3	0–3	Indicates the group ID of de-interlace operation parameters
t1	0–2	Indicates the number of low-luminance lookup table data segments
t2	0–2	Indicates the number of medium-luminance lookup table data segments
t3	0–2	Indicates the number of high-luminance lookup table data segments
t4	0–2	Indicates the number of middle_low-luminance lookup table data segments
t5	0–2	Indicates the number of middle_high-luminance lookup table data segments
m	0–31	Indicates the number of cascaded subimages

6.2.7 Register Description

VO_CTRL

VO_CTRL is the VO control register.

Offset Address	Register Name	Total Reset Value
0x0000	VO_CTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												outstd_wid0				reserved				outstd_rid0				reserved				arb_mode			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:20]	-	reserved	Reserved.
[19:16]	RW	outstd_wid0	Outstanding of ID0 written by the AXI bus.
[15:12]	-	reserved	Reserved.
[11:8]	RW	outstd_rid0	Outstanding of ID0 read by the AXI bus.
[7:4]	-	reserved	Reserved.
[3:0]	-	arb_mode	Arbitration mode of the data requests from each internal surface bus of the VOU. 0x0: polling mode 0x1: graphics layers first Others: reserved

VO_INTSTA

VO_INTSTA is the VO interrupt status register. Writing 1 to an interrupt status bit clears the corresponding interrupt.

Offset Address	Register Name	Total Reset Value
0x0004	VO_INTSTA	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	be_int	reserved	vhd_st_wr_int	resereved	hcrr_int	resereved	g2rr_int	g1rr_int	g0rr_int	vhdrr_int	resereved	vsdrr_int	reserved														dhduf_int	dhdvtthd_int	resereved		dsduf_int	dsdvtthd_int
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	WC	be_int	Bus (AXI_Master) error interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[30]	-	reserved	Reserved.
[29]	WC	vhd_st_wr_int	VHD write bus low-bandwidth interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[28]	-	reserved	Reserved.
[27]	WC	hcrr_int	HC register update complete interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[26]	-	reserved	Reserved.
[25]	WC	g2rr_int	G2 register update complete interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[24]	WC	g1rr_int	G1 register update complete interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[23]	WC	g0rr_int	G0 register update complete interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[22]	WC	vhdrr_int	VDC_HD register update complete interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[21]	-	reserved	Reserved.

[20]	WC	vsdrr_int	VDC_SD register update complete interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[19:6]	-	reserved	Reserved.
[5]	WC	dhduf_int	Low-bandwidth alarm interrupt status of the HD channel. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	WC	dhdvtthd_int	Vertical timing interrupt status of the HD channel. 0: No interrupt is generated. 1: An interrupt is generated.
[3]	-	reserved	Reserved.
[2]	-	reserved	Reserved.
[1]	WC	dsduf_int	Low-bandwidth alarm interrupt status of the SD channel. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	WC	dsdvtthd_int	Vertical timing interrupt status of the SD channel. 0: No interrupt is generated. 1: An interrupt is generated.

VO_INTMSK

VO_INTMSK is the VO interrupt mask register. It is related to VO_INTSTA.

Offset Address	Register Name	Total Reset Value
0x0008	VO_INTMSK	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	be_intmsk	reserved	vhd_st_wr_intmask	resereved	hcrr_intmsk	resereved	g2rr_intmsk	g1rr_intmsk	g0rr_intmsk	resereved	vadrr_intmsk	vsdrr_intmsk	reserved														dhduf_intmsk	dhdvtthd_intmsk	resereved		dsduf_intmsk	dsdvtthd_intmsk
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	be_intmsk	Bus (AXI_Master) error interrupt mask. 0: masked 1: not masked

[30]	-	reserved	Reserved.
[29]	RW	vhd_st_wr_intmask	VHD write bus low-bandwidth interrupt mask. 0: masked 1: not masked
[28]	-	reserved	Reserved.
[27]	RW	hcrr_intmsk	HC register update complete interrupt mask. 0: masked 1: not masked
[26]	-	reserved	Reserved.
[25]	RW	g2rr_intmsk	G2 register update complete interrupt mask. 0: masked 1: not masked
[24]	RW	g1rr_intmsk	G1 register update complete interrupt mask. 0: masked 1: not masked
[23]	RW	g0rr_intmsk	G0 register update complete interrupt mask. 0: masked 1: not masked
[22]	RW	vhdrr_intmsk	VDC_HD register update complete interrupt mask. 0: masked 1: not masked
[21]	-	reserved	Reserved.
[20]	RW	vsdrr_intmsk	VDC_SD register update complete interrupt mask. 0: masked 1: not masked
[19:6]	-	reserved	Reserved.
[5]	RW	dhduf_intmsk	Low-bandwidth alarm interrupt mask of the HD channel. 0: masked 1: not masked
[4]	RW	dhdvtthd_intmsk	Timing threshold interrupt mask of the HD channel. 0: masked 1: not masked
[3]	-	reserved	Reserved.
[2]	-	reserved	Reserved.
[1]	RW	dsduf_intmsk	Low-bandwidth alarm interrupt mask of the SD channel. 0: masked 1: not masked

[0]	RW	dsdvtthd_intmsk	Timing threshold interrupt mask of the SD channel. 0: masked 1: not masked

VO_VERSION1

VO_VERSION1 is VO version register 1.

Offset Address	Register Name	Total Reset Value
0x000C	VO_VERSION1	0x7675_6F76

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	voversion0																															
Reset	0	1	1	1	0	1	1	0	0	1	1	1	0	1	0	1	0	1	1	0	1	1	1	1	0	1	1	1	0	1	1	0

Bits	Access	Name	Description
[31:0]	RO	voversion0	VO version register.

VO_VERSION2

VO_VERSION2 is VO version register 2.

Offset Address	Register Name	Total Reset Value
0x0010	VO_VERSION2	0x3031_3034

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	voversion1																															
Reset	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0

Bits	Access	Name	Description
[31:0]	RO	voversion1	VO version register.

VO_PARAUP

VO_PARAUP is the scaling and gamma coefficient update enable register. The VO scaling coefficients are configured by the AXI master. The flags, which indicate whether the start addresses and parameters need to be updated, are configured by the APB slave.

Offset Address	Register Name	Total Reset Value
0x002C	VO_PARAUP	0x0000_0000

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

Bits	Access	Name	Description
[31:6]	-	reserved	Reserved.
[5]	-	reserved	Reserved.
[4]	RW	dhd_acc_upd	Indicates whether the DHD ACC operation lookup table needs to be updated. This bit is cleared automatically after the hardware updates the lookup table. 0: not updated 1: updated
[3]	-	reserved	Reserved.
[2]	-	reserved	Reserved.
[1]	RW	vhd_vcoef_upd	Indicates whether the VHD vertical luminance and chrominance filtering coefficients need to be updated. This bit is cleared automatically after the hardware updates the coefficients. 0: not updated 1: updated
[0]	RW	vhd_hcoef_upd	Indicates whether the VHD horizontal luminance and chrominance filtering coefficients need to be updated. This bit is cleared automatically after the hardware updates the coefficients. 0: not updated 1: updated

VHDHCOEFAD

VHDHCOEFAD is the horizontal luminance and chrominance filtering coefficient address register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x0030	VHDHCOEFAD	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	coef_addr	Start address of the coefficients stored in the local memory.

VHDVCOEFAD

VHDVCOEFAD is the vertical luminance and horizontal chrominance filtering coefficient address register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x0034	VHDVCOEFAD	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	coef_addr	Start address of the coefficients stored in the local memory.

DHDACCAD

DHDACCAD is the ACC coefficient lookup table address register of the DHD channel.

Offset Address	Register Name	Total Reset Value
0x0040	DHDACCAD	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	coef_addr	Start address of the coefficients stored in the local memory.

VHDCTRL

VHDCTRL is the VHD channel control register.

Offset Address	Register Name	Total Reset Value
0x0100	VHDCTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vhd_en	reserved													regup_rate	bfield_first	lm_rmode		chm_rmode		reserved								ifmt			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	vhd_en	Surface enable. 0: disabled 1: enabled
[30:18]	-	reserved	Reserved.
[17]	RW	regup_rate	Frequency of updating surface registers in interlaced output mode. 0: once for a frame 1: once for a field
[16]	RW	bfield_first	Bottom field first. 0: top field first, T0B0T1B1 (T0B0 is regarded as a frame) 1: bottom field first, B0T0B1T1 (B0T0 is regarded as a frame) Note: The letter *T* indicates top field and the letter *B* indicates bottom field.
[15:14]	RW	lm_rmode	Luminance read mode. 00: Reads the frame data buffer in interlaced mode and reads the top fields and bottom fields based on the timing sequence. 01: Reads the frame data buffer in progressive mode. 10: Reads only top fields when each frame is displayed in interlaced mode. 11: Reads only bottom fields when each frame is displayed in interlaced mode.
[13:12]	RW	chm_rmode	Chrominance read mode. 00: Reads the frame data buffer in interlaced mode and reads the top fields and bottom fields based on the timing sequence. 01: Reads the frame data buffer in progressive mode. 10: Reads only top fields when each frame is displayed in interlaced mode. 11: Reads only bottom fields when each frame is displayed in interlaced mode.

[11:4]	-	reserved	Reserved.
[3:0]	RW	ifmt	Input data format. 0x3: SPYCbCr4:2:0 0x4: SPYCbCr4:2:2 (in the format of 1x2) **Others: reserved**

VHDUPD

VHDUPD is the VHD channel update enable register.

Offset Address	Register Name	Total Reset Value
0x0104	VHDUPD	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	regup	Register update bit of surfaces. After the registers at the HD video layer are configured, the registers are updated when the value 1 is written to this bit. After updates, this bit is cleared automatically by the hardware.

VHDLADDR

VHDLADDR is the address register of the previous de-interlace frame of the VHD channel For the package pixel, the address is the address of the frame buffer; for the semi-planar pixel, the address is the address of the luminance frame buffer.

Offset Address	Register Name	Total Reset Value
0x0108	VHDLADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vhdladdr	Address of the previous de-interlace frame.

VHDLCADDR

VHDLCADDR is the chrominance address register of the previous de-interlace frame of the VHD channel. For the package pixel, the address is invalid; for the semi-planar pixel, the address is the address of the luminance frame buffer.

Offset Address	Register Name	Total Reset Value
0x010C	VHDLCADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vhdlcaddr	Chrominance address of the previous de-interlace frame.

VHDCADDR

VHDCADDR is the address register of the current de-interlace frame of the VHD channel. For the package pixel, the address is the address of the frame buffer; for the semi-planar pixel, the address is the address of the luminance frame buffer.

Offset Address	Register Name	Total Reset Value
0x0110	VHDCADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vhdcaddr	Address of the current de-interlace frame.

VHDCCADDR

VHDCCADDR is the chrominance address register of the current de-interlace frame of the VHD channel. For the package pixel, the address is invalid; for the semi-planar pixel, the address is the address of the luminance frame buffer.

Offset Address	Register Name	Total Reset Value
0x0114	VHDCCADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vhdccaddr	Chrominance address of the current de-interlace frame.

VHDNADDR

VHDNADDR is the address register of the next de-interlace frame of the VHD channel. For the package pixel, the address is the address of the frame buffer; for the semi-planar pixel, the address is the address of the luminance frame buffer.

Offset Address	Register Name	Total Reset Value
0x0118	VHDNADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vhdnaddr	Address of the next de-interlace frame.

VHDNCADDR

VHDNCADDR is the chrominance address register of the current de-interlace frame of the VHD channel. For the package pixel, the address is invalid; for the semi-planar pixel, the address is the address of the luminance frame buffer.

Offset Address	Register Name	Total Reset Value
0x011C	VHDNCADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vhdncaddr	Chrominance address of the next de-interlace frame.

VHDSTRIDE

VHDSTRIDE is the line stride register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x0120	VHDSTRIDE	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RW	vhdcstride	Stride of the chrominance frame buffer (in the format of semi-planar), in the unit of word.
[15:0]	RW	vhdstride	Stride of the luminance frame buffer (in the format of semi-planar), in the unit of word.

VHDCBMPARA

VHDCBMPARA is the blending parameter register of the VHD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0124	VHDCBMPARA	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	-	reserved	Reserved.
[7:0]	RW	galpha	Global blending alpha value. This value ranges from 0 to 127. The value 0 indicates full transparent and the value 127 indicates opaque.

VHDORESO

VHDORESO is the output resolution register of the VHD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0128	VHDORESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	oh	Height of an output video image, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of an output image at a video layer must be an even number. There is no such restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	ow	Width of an output video image, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of an output image at a video layer must be an even number.

VHDIRESO

VHDIRESO is the input resolution register of the VHD channel. This register is a non-instant register.

CAUTION

- In interlaced output mode, the actual height of an input image must be an even number. When the input data format is YCbCr4:2:0, the image height must be an integer multiple of 4.
- In progressive output mode, when the input data format is YCbCr4:2:0, the image height must be an integer multiple of 2. In addition, the actual width of an input image at a video layer must be an even number.

Offset Address	Register Name	Total Reset Value
0x012C	VHDIRESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	ih	Height of an input video image, in the unit of row. The configured value is equal to the actual height minus 1.
[15:12]	-	reserved	Reserved.
[11:0]	RW	iw	Width of an input video image, in the unit of pixel. The configured value is equal to the actual width minus 1.

VHDSFPOS

VHDSFPOS is the source bitmap start position register of the VHD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0130	VHDSFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	src_yfpos	Y coordinate of the source start coordinates. The value 0 indicates the first row. In interlaced output mode, the configured value must be an even number.
[15:12]	-	reserved	Reserved.
[11:0]	RW	src_xfpos	X coordinate of the source start coordinates. The value 0 indicates the first pixel in the first row.

VHDDFPOS

VHDDFPOS is the display window start position register of the VHD channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0134	VHDDFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	disp_yfpos	Start coordinates of the display column.
[15:12]	RW	reserved	Reserved.
[11:0]	RW	disp_xfpos	Start coordinates of the display row.

VHDDLPOS

VHDDLPOS is the display window end position register of the VHD channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0138	VHDDLPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	disp_ylpos	End coordinates of the display column.
[15:12]	-	reserved	Reserved.
[11:0]	RW	disp_xlpos	End coordinates of the display row.

VHDVFPOS

VHDVFPOS is the video start position register in the display window of the VHD channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x013C	VHDVFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	video_yfpos	Start coordinates of the video column.
[15:12]	-	reserved	Reserved.
[11:0]	RW	video_xfpos	Start coordinates of the video row.

VHDVLPOS

VHDVLPOS is the video end position register in the display window of the VHD channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0140	VHDVLPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	video_ylpos	End coordinates of the video column.
[15:12]	-	reserved	Reserved.
[11:0]	RW	video_xlpos	End coordinates of the video row.

VHDBK

VHDBK is the video layer background color register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x0144	VHDBK	0x0000_0000

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

Bits	Access	Name	Description
[31:24]	RW	vbk_alpha	Background filling color of a video layer. Its value ranges from 0 to 128.
[23:16]	RW	vbk_y	Component Y
[15:8]	RW	vbk_cb	Component Cb
[7:0]	RW	vbk_cr	Component Cr

VHDLMSP

VHDLMSP is the luminance scaling parameter configuration register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x0150	VHDLMSP	0x0000_0000

Bit	31	30	29	28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	hlmsc_en	vlmsc_en	hlmid_en	vlmid_en	reserved		shift_field	fld_offset			
Reset	0	0	0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31]	RW	hlmsc_en	Luminance horizontal scaling enable. 0: disabled 1: enabled
[30]	RW	vlmsc_en	Luminance vertical scaling enable. 0: disabled 1: enabled
[29]	RW	hlmid_en	Luminance horizontal scaling and median filtering enable. 0: disabled 1: enabled

[28]	RW	vlmid_en	Luminance vertical scaling and median filtering enable. 0: disabled 1: enabled
[27:17]	-	reserved	Reserved.
[16]	RW	shift_field	Field affected due to shift. 0: bottom field 1: top field
[15:0]	RW	fld_offset	Field shift. The value is expressed as a complementary code and is in the format of 4.12. The most significant bit (MSB) is the sign bit. Note: This value is related to the scaling ratio.

VHDCHMSP

VHDCHMSP is the chrominance scaling parameter configuration register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x0154	VHDCHMSP	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	hchmsc_en	vchmsc_en	hlmid_en	vlmid_en	reserved											shift_field	fld_offset															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	hchmsc_en	Chrominance horizontal scaling enable. 0: disabled 1: enabled
[30]	RW	vchmsc_en	Chrominance vertical scaling enable. 0: disabled 1: enabled
[29]	RW	hlmid_en	Chrominance horizontal scaling and median filtering enable. 0: disabled 1: enabled
[28]	RW	vlmid_en	Chrominance vertical scaling and median filtering enable. 0: disabled 1: enabled

[27:17]	-	reserved	Reserved.
[16]	RW	shift_field	Field affected due to shift. 0: bottom field 1: top field
[15:0]	RW	fld_offset	Field shift. The value is expressed as a complementary code and is in the format of 4.12. The MSB is the sign bit. Note: This value is related to the scaling ratio.

VHDLMHSP

VHDLMHSP is the luminance horizontal scaling parameter configuration register of the VHD channel. This register is a non-instant register.

Horizontal scaling ratio of the luminance = input image width/output image width

Offset Address	Register Name	Total Reset Value
0x0158	VHDLMHSP	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	lm_hphase	Horizontal scaling initial phase of the luminance, in the format of 0.12.
[15:0]	RW	lm_hratio	Horizontal scaling ratio of the luminance, in the format of 4.12.

VHDLMVSP

VHDLMVSP is the luminance vertical scaling parameter configuration register of the VHD channel. This register is a non-instant register.

Vertical scaling ratio of the luminance = input image height/output image height

Offset Address	Register Name	Total Reset Value
0x015C	VHDLMVSP	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	lm_vphase	Vertical scaling initial phase of the luminance, in the format of 0.12.
[15:0]	RW	lm_vratio	Vertical scaling ratio of the luminance, in the format of 4.12.

VHDCHMHSP

VHDCHMHSP is the chrominance horizontal scaling parameter configuration register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x0160	VHDCHMHSP	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	chm_hphase	Horizontal scaling initial phase of the chrominance, in the format of 0.12
[15:0]	RW	chm_hratio	Horizontal scaling ratio of the chrominance, in the format of 4.12.

VHDCHMVSP

VHDCHMVSP is the chrominance vertical scaling parameter configuration register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x0164	VHDCHMVSP	0x0000_0000

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

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	chm_hphase	Vertical scaling initial phase of the chrominance, in the format of 1.12. The MSB is a sign bit.
[15:0]	RW	chm_vratio	Vertical scaling ratio of the chrominance, in the format of 4.12.

VHDDIECTRL

VHDDIECTRL is the de-interlace operation control register of the VHD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0170	VHDDIECTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	die_luma_en	die_chroma_en	reserved	die_frt	reserved	die_lmmode	reserved	die_chmmode	die_rf_mode	reserved	lm_mov_tsmix_en	lm_st_tsmix_en	lm_tflt_en	chm_tsmix_en	stinfo_rst	reserved															die_reff_cfg_en	die_reff_cfg
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	die_luma_en	De-interlace luminance enable. 0: disabled 1: enabled
[30]	RW	die_chroma_en	De-interlace chrominance enable. 0: disabled 1: enabled
[29]	-	reserved	Reserved.

[28]	RW	die_frt	De-interlace ratio of the input field rate to the output frame rate. 0: input field rate = output frame rate 1: input field rate = 2 x output frame rate
[27]	-	reserved	Reserved.
[26]	RW	die_lmmode	De-interlace luminance operation mode. 0: 4-field mode 1: 2-field median mode
[25]	-	reserved	Reserved.
[24]	RW	die_chmmode	De-interlace chrominance operation mode. 0: 4-field mode 1: 2-field median mode
[23]	RW	die_rf_mode	Bandwidth-saving control for the reference field in de-interlace 4-field mode. 0: The reference field reads the data of the full field. 1: The reference field reads the data of half the field.
[22]	-	reserved	Reserved.
[21]	RW	lm_mov_tsmix_en	Luminance spatio-temporal weight enable (motion part). 0: disabled 1: enabled
[20]	RW	lm_st_tsmix_en	Luminance spatio-temporal weight enable (still part). 0: disabled 1: enabled
[19]	RW	lm_tflt_en	Luminance time-domain filtering enable. 0: disabled 1: enabled
[18]	RW	chm_tsmix_en	Chrominance spatio-temporal weight enable. 0: disabled 1: enabled
[17]	RW	stinfo_rst	Still times reset control. 0: The still times are updated properly. 1: The still times are cleared.
[16]	-	reserved	Reserved.
[15:2]	-	reserved	Reserved.

[1]	RW	die_reff_cfg_en	Reference field configuration control. 0: The reference fields configured by the hardware are used during de-interlace. 1: The reference fields configured by the software are used during de-interlace, that is, the die_reff_cfg configuration is valid. Note: In de-interlace mode, this bit is valid when VHDCTRL[regup_date] is set to 1.
[0]	RW	die_reff_cfg	Indicates which field in the reference fields configured by the software is de-interlaced (reserved field). 0: The top field is reserved. 1: The bottom field is reserved. Note: This bit is valid when die_reff_cfg_en is set to 1.

VHDDIETHD

VHDDIETHD is the de-interlace operation threshold register of the VHD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0174	VHDDIETHD	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												fld_diff_thd				med_thd				reserved	st_thd			reserved				md_thd			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	-	reserved	Reserved.
[21:16]	RW	fld_diff_thd	Field differential threshold, for selecting the coefficient of the spatio-temporal weight.
[15:12]	RW	med_thd	Median detection threshold.
[11]	-	reserved	Reserved.
[10:8]	RW	st_thd	Still times threshold.
[7:5]	-	reserved	Reserved.
[4:0]	RW	md_thd	Motion/still decision threshold.

VHDDIEADDR

VHDDIEADDR is the de-interlace history buffer address register of the VHD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0178	VHDDIEADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	dieaddr	Buffer address for storing the de-interlace history information.

VHDDIETSMIX

VHDDIETSMIX is the de-interlace spatio-temporal weight coefficient register of the VHD channel. This register is an instant register. There are four groups of spatio-temporal weight coefficients and each group consists of eight coefficients.

Offset Address	Register Name	Total Reset Value
0x0180+n1×0x4	VHDDIETSMIX	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	tsmix_n7				tsmix_n6				tsmix_n5				tsmix_n4				tsmix_n3				tsmix_n2				tsmix_n1				tsmix_n0			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:28]	RW	tsmix_n7	Spatio-temporal weight coefficient 7 of group n1 (n1 = 0–3).
[27:24]	RW	tsmix_n6	Spatio-temporal weight coefficient 6 of group n1 (n1 = 0–3).
[23:20]	RW	tsmix_n5	Spatio-temporal weight coefficient 5 of group n1 (n1 = 0–3).
[19:16]	RW	tsmix_n4	Spatio-temporal weight coefficient 4 of group n1 (n1 = 0–3).
[15:12]	RW	tsmix_n3	Spatio-temporal weight coefficient 3 of group n1 (n1 = 0–3).
[11:8]	RW	tsmix_n2	Spatio-temporal weight coefficient 2 of group n1 (n1 = 0–3).
[7:4]	RW	tsmix_n1	Spatio-temporal weight coefficient 1 of group n1 (n1 = 0–3).
[3:0]	RW	tsmix_n0	Spatio-temporal weight coefficient 0 of group n1 (n1 = 0–3).

VHDDIETFLT

VHDDIETFLT is the de-interlace time-domain filtering coefficient register of the VHD channel. This register is an instant register. There are four groups of time-domain filtering coefficients and each group consists of eight coefficients.

Offset Address	Register Name	Total Reset Value
0x0190+n2×0x4	VHDDIETFLT	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	tflt_n7	tflt_n6	tflt_n5	tflt_n4	tflt_n3	tflt_n2	tflt_n1	tflt_n0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:28]	RW	tflt_n7	Time-domain filtering coefficient 7 of group n2 (n2 = 0–3).
[27:24]	RW	tflt_n6	Time-domain filtering coefficient 6 of group n2 (n2 = 0–3).
[23:20]	RW	tflt_n5	Time-domain filtering coefficient 5 of group n2 (n2 = 0–3).
[19:16]	RW	tflt_n4	Time-domain filtering coefficient 4 of group n2 (n2 = 0–3).
[15:12]	RW	tflt_n3	Time-domain filtering coefficient 3 of group n2 (n2 = 0–3).
[11:8]	RW	tflt_n2	Time-domain filtering coefficient 2 of group n2 (n2 = 0–3).
[7:4]	RW	tflt_n1	Time-domain filtering coefficient 1 of group n2 (n2 = 0–3).
[3:0]	RW	tflt_n0	Time-domain filtering coefficient 0 of group n2 (n2 = 0–3).

VHDDIEVFLT

VHDDIEVFLT is the de-interlace vertical filtering weight coefficient register of the VHD channel. This register t is an instant register. There are four groups of operation parameters and each group consists of eight coefficients.

Offset Address	Register Name	Total Reset Value
0x01A0+n3×0x4	VHDDIEVFLT	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	vflt_n7	vflt_n6	vflt_n5	vflt_n4	vflt_n3	vflt_n2	vflt_n1	vflt_n0
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:28]	RW	vflt_n7	De-interlace vertical filtering weight coefficient 7 of group n3 (n3 = 0–3).
[27:24]	RW	vflt_n6	De-interlace vertical filtering weight coefficient 6 of group n3 (n3 = 0–3).
[23:20]	RW	vflt_n5	De-interlace vertical filtering weight coefficient 5 of group n3 (n3 = 0–3).
[19:16]	RW	vflt_n4	De-interlace vertical filtering weight coefficient 4 of group n3 (n3 = 0–3).

[15:12]	RW	vflt_n3	De-interlace vertical filtering weight coefficient 3 of group n3 (n3 = 0–3).
[11:8]	RW	vflt_n2	De-interlace vertical filtering weight coefficient 2 of group n3 (n3 = 0–3).
[7:4]	RW	vflt_n1	De-interlace vertical filtering weight coefficient 1 of group n3 (n3 = 0–3).
[3:0]	RW	vflt_n0	De-interlace vertical filtering weight coefficient 0 of group n3 (n3 = 0–3).

VHDSTATUS

VHDSTATUS is the video layer status register of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x01F0	VHDSTATUS	0x0800_0001

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RO	die_ref_field	Reference field of the de-interlace operation. When the de-interlace function is disabled, the value of this register is 0. 0: The top field is referenced. 1: The bottom field is referenced.

VSDCTRL

VSDCTRL is the VSD channel control register.

Offset Address	Register Name	Total Reset Value
0x0300	VSDCTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vsd_en	reserved													regup_rate	bfield_first	lm_rmode		chm_rmode		reserved								ifmt			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	vsd_en	Surface enable. 0: disabled 1: enabled
[30:18]	-	reserved	Reserved.
[17]	RW	regup_rate	Frequency of updating surface registers in interlaced output mode. 0: once for a frame 1: once for a field
[16]	RW	bfield_first	Bottom field first. 0: top field first, T0B0T1B1 (T0B0 is regarded as a frame) 1: bottom field first, B0T0B1T1 (B0T0 is regarded as a frame)
[15:14]	RW	lm_rmode	Luminance read mode. 00: Reads the frame data buffer in interlaced mode and reads the top fields and bottom fields based on the timing sequence. 01: Reads the frame data buffer in progressive mode. 10: Reads only top fields when each frame is displayed in interlaced mode. 11: Reads only bottom fields when each frame is displayed in interlaced mode.
[13:12]	RW	chm_rmode	Chrominance read mode. 00: Reads the frame data buffer in interlaced mode and reads the top fields and bottom fields based on the timing sequence. 01: Reads the frame data buffer in progressive mode. 10: Reads only top fields when each frame is displayed in interlaced mode. 11: Reads only bottom fields when each frame is displayed in interlaced mode.
[11:4]	-	reserved	Reserved.
[3:0]	RW	ifmt	Input data format. 0x3: SPYCbCr4:2:0 0x4: SPYCbCr4:2:2 (in the format of 1x2) Others: reserved

VSDUPD

VSDUPD is the VSD channel update enable register.

Offset Address	Register Name	Total Reset Value
0x0304	VSDUPD	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	regup	Register update bit of surfaces. After the registers at the SD video layer are configured, the registers are updated when the value 1 is written to this bit. After updates, this bit is cleared automatically by the hardware.

VSDADDR

VSDADDR is the address register of the current frame of the VSD channel For the package pixel, the address is the address of the frame buffer; for the semi-planar pixel, the address is the address of the luminance frame buffer.

Offset Address	Register Name	Total Reset Value
0x0310	VSDADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vsdaddr	Buffer address for storing surface frames.

VSDCADDR

VSDCADDR is the chrominance address register of the current frame of the VSD channel. For the package pixel, the address is invalid; for the semi-planar pixel, the address is the address of the luminance frame buffer.

Offset Address	Register Name	Total Reset Value
0x0314	VSDCADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	vsdcaddr	Address of the surface chrominance frame buffer, in the unit of word.

VSDSTRIDE

VSDSTRIDE is the line stride register of the VSD channel.

Offset Address	Register Name	Total Reset Value
0x0320	VSDSTRIDE	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RW	vsdcstride	Stride of the chrominance frame buffer (in the format of semi-planar), in the unit of word.
[15:0]	RW	vsdstride	Stride of the luminance frame buffer (in the format of semi-planar), in the unit of word.

VSDCBMPARA

VSDCBMPARA is the blending parameter register of the VSD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0324	VSDCBMPARA	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	-	reserved	Reserved.
[7:0]	RW	galpha	Global blending alpha value. This value ranges from 0 to 127. The value 0 indicates full transparent and the value 127 indicates opaque.

VSDORESO

VSDORESO is the output resolution register of the VSD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0328	VSDORESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	oh	Height of an output video image, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of an output image at a video layer must be an even number. There is no restriction in progressive mode.
[15:12]	RW	reserved	Reserved.
[11:0]	RW	ow	Width of an output video image, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of an output image at a video layer must be an even number.

VSDIRESO

VSDIRESO is the input resolution register of the VSD channel. This register is a non-instant register.

CAUTION

- In interlaced output mode, the actual height of an input image must be an even number. When the input data format is YCbCr4:2:0, the height must an integer multiple of 4.
- In progressive output mode, when the input data format is YCbCr4:2:0, the image height must be an integer multiple of 2. In addition, the actual image width of an input image at a video layer must be an even number.

Offset Address	Register Name	Total Reset Value
0x032C	VSDIRESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	ih	Height of an input video image, in the unit of row. The configured value is equal to the actual height minus 1.
[15:12]	-	reserved	Reserved.
[11:0]	RW	iw	Width of an input video image, in the unit of pixel. The configured value is equal to the actual width minus 1.

VSDSFPOS

VSDSFPOS is the source bitmap start position register of the VSD channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0330	VSDSFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	src_yfpos	Y coordinate of the source start coordinates. The value 0 indicates the first row. In interlaced output mode, the configured value must be an even number.
[15:12]	RW	reserved	Reserved.
[11:0]	RW	src_xfpos	X coordinate of the source start coordinates. The value 0 indicates the first pixel in the first row.

VSDDFPOS

VSDDFPOS is the display window start position register of the VSD channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0334	VSDDFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	disp_yfpos	Start coordinates of the display column.
[15:12]	-	reserved	Reserved.
[11:0]	RW	disp_xfpos	Start coordinates of the display row.

VSDDLPOS

VSDDLPO is the display window end position register of the VSD channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0338	VSDDLPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	disp_ylpos	End coordinates of the display column.
[15:12]	-	reserved	Reserved.
[11:0]	RW	disp_xlpos	End coordinates of the display row.

VSDVFPOS

VSDVFPOS is the video start position register in the display window of the VSD channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x033C	VSDVFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	video_yfpos	Start coordinates of the video column.
[15:12]	-	reserved	Reserved.
[11:0]	RW	video_xfpos	Start coordinates of the video row.

VSDVLPOS

VSDVLPOS is the video end position register in the display window of the VSD channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0340	VSDVLPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	video_ylpos	End coordinates of the video column.
[15:12]	-	reserved	Reserved.
[11:0]	RW	video_xlpos	End coordinates of the video row.

VSDBK

VSDBK is the video layer background color register of the VSD channel.

Offset Address	Register Name	Total Reset Value
0x0344	VSDBK	0x0000_0000

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

Bits	Access	Name	Description
[31:24]	RW	vbk_alpha	Background filling color of a video layer. Its value ranges from 0 to 128.
[23:16]	RW	vbk_y	Component Y.
[15:8]	RW	vbk_cb	Component Cb.
[7:0]	RW	vbk_cr	Component Cr.

G0CTRL

G0CTRL is the G0 channel control register. It is a non-instant register. It is used to configure the information about graphics layers.

Offset Address	Register Name	Total Reset Value
0x0400	G0CTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	g0_en	csc_en	csc_mo d	p r e	resereved																		bitext		ifmt							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	g0_en	Surface enable. 0: disabled 1: enabled
[30]	RW	csc_en	Color space conversion enable. 0: disabled 1: enabled
[29]	RW	csc_mode	Color space conversion standard. 0: BT.601 1: BT.709
[28]	RW	pre_en	Data format premultiplied enable. 0: disabled 1: enabled
[27:10]	-	reserved	Reserved.
[9:8]	RW	bitext	Bit extend mode of the surface input bitmap. 0X: extend 0s to lower bits 10: extend the value of the MSB to lower bits 11: extend the values of upper bits to lower bits
[7:0]	RW	ifmt	Input data format. 0x49: aRGB1555 Others: reserved

G0UPD

G0UPD is the G0 channel update enable register.

Offset Address	Register Name	Total Reset Value
0x0404	G0UPD	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	regup	Register update bit of surfaces. After the registers at graphics layer 0 are configured, the registers are updated when the value 1 is written to this bit. After updates, this bit is cleared automatically by the hardware.

G0ADDR

G0ADDR is the frame address register of the G0 channel.

Offset Address	Register Name	Total Reset Value
0x0408	G0ADDR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	g0addr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	g0addr	Buffer address for storing surface frames.

G0STRIDE

G0STRIDE is the frame stride register of the G0 channel.

Offset Address	Register Name	Total Reset Value
0x040C	G0STRIDE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																g0stride															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RW	g0stride	Row stride of the frame buffer.

G0CBMPARA

G0CBMPARA is the blending parameter register of the G0 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0410	G0CBMPARA	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																key_mode	key_en	reserved	palpha_en	reserved			palpha_range	galpha							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15]	RW	key_mode	Colorkey mode. 0: If Keymin ≤ Pixel ≤ Keymax, the color is regarded as the colorkey. 1: If Pixel ≤ Keymin or Pixel ≥ Keymax, the color is regarded as the colorkey.
[14]	RW	key_en	Colorkey enable. 0: disabled 1: enabled
[13]	-	reserved	Reserved.
[12]	RW	palpha_en	Pixel alpha enable. 0: disabled 1: enabled

[11:9]	-	reserved	Reserved.
[8]	RW	palpha_range	Range of the pixel alpha. 0: 0–128 1: 0–255
[7:0]	RW	galpha	Global blending alpha value. This value ranges from 0 to 127. The value 0 indicates full transparent and the value 127 indicates opaque.

G0CKEYMAX

G0CKEYMAX is the maximum colorkey value register of the G0 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0414	G0CKEYMAX	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	va0		keyr_max		keyg_max		keyb_max	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:24]	RW	va0	Alpha0 value. When the data format is alpha RGB1555 and the value of the MSB is 0, the alpha value is replaced with the value of alpha0.
[23:16]	RW	keyr_max	Maximum value of component R of the colorkey.
[15:8]	RW	keyg_max	Maximum value of component G of the colorkey.
[7:0]	RW	keyb_max	Maximum value of component B of the colorkey.

G0CKEYMIN

G0CKEYMIN is the minimum colorkey value register of the G0 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0418	G0CKEYMIN	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	va1		keyr_min		keyg_min		keyb_min	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:24]	RW	va1	Alpha1 value. When the data format is alpha RGB1555 and the value of the MSB is 0, the alpha value is replaced with the value of alpha1.
[23:16]	RW	keyr_min	Minimum value of component R of the colorkey.
[15:8]	RW	keyg_min	Minimum value of component G of the colorkey.
[7:0]	RW	keyb_min	Minimum value of component B of the colorkey.

G0IRESO

G0IRESO is the input resolution register of the G0 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x041C	G0IRESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	ih	Height of an input image at a graphics layer, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of an input image at a graphics layer must be an even number. There is no such restriction in progressive output mode
[15:12]	-	reserved	Reserved.
[11:0]	RW	iw	Width of an input image at a graphics layer, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of an input image at a graphics layer must be an even number.

G0ORESO

G0ORESO is the output resolution register of the G0 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0420	G0ORESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	oh	Height of an output image at a graphics layer, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of an output image at a graphics layer must be an even number. There is no such restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	ow	Width of an output image at a graphics layer, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of an output image at a graphics layer must be an even number.

G0SFPOS

G0SFPOS is the source bitmap start position register of the G0 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0424	G0SFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	src_yfpos	Y coordinate of the source start coordinates. The value 0 indicates the first row. In interlaced output mode, the configured value must be an even number.

[15:12]	-	reserved	Reserved.
[11:0]	RW	src_xfpos	X coordinate of the source start coordinates. The value 0 indicates the first pixel in the first row.

G0DFPOS

G0DFPOS is the display window start position register of the G0 channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0428	G0DFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	disp_yfpos	Column start coordinates.
[15:12]	-	reserved	Reserved.
[11:0]	RW	disp_xfpos	Row start coordinates.

G0DLPOS

G0DLPOS is the display window end position register of the G0 channel (in pixels). This register is a non-instant register.

Offset Addresss0x042C	Register Name	Total Reset Value
	G0DLPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	disp_ylpos	Column end coordinates.
[15:12]	-	reserved	Reserved.
[11:0]	RW	disp_xlpos	Row end coordinates.

G1CTRL

G1CTRL is the G1 channel control register. It is a non-instant register. It is used to configure the information about graphics layers.

Offset Address	Register Name	Total Reset Value
0x0500	G1CTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	g1_en	csc_en	csc_mode	pre	resereved																		bitext		ifmt							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	g1_en	Surface enable. 0: disabled 1: enabled
[30]	RW	csc_en	Color space conversion enable. 0: disabled 1: enabled
[29]	RW	csc_mode	Color space conversion standard. 0: BT.601 1: BT.709
[28]	RW	pre_en	Data format premultiplied enable. 0: disabled 1: enabled
[27:10]	-	reserved	Reserved.
[9:8]	RW	bitext	Bit extend mode of the surface input bitmap. 0X: extend 0s to lower bits 10: extend the value of the MSB to a lower bit 11: extend the values of upper bits to lower bits
[7:0]	RW	ifmt	Input data format. 0x49: aRGB1555 Others: reserved

G1UPD

G1UPD is the G1 channel update enable register.

Offset Address	Register Name	Total Reset Value
0x0504	G1UPD	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	regup	Register update bit of surfaces. After the registers at graphics layer 1 are configured, the registers are updated when the value 1 is written to this bit. After updates, this bit is cleared automatically by the hardware.

G1ADDR

G1ADDR is the frame address register of the G1 channel.

Offset Address	Register Name	Total Reset Value
0x0508	G1ADDR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	g1addr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	g1addr	Buffer address for storing surface frames.

G1STRIDE

G1STRIDE is the frame stride register of the G1 channel.

Offset Address	Register Name	Total Reset Value
0x050C	G1STRIDE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																g1stride															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RW	g1stride	Row stride of the frame buffer.

G1CBMPARA

G1CBMPARA is the blending parameter register of the G1 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0510	G1CBMPARA	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																key_mode	key_en	reserved	palpha_en	reserved			palpha_range	galpha								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15]	RW	key_mode	Colorkey mode. 0: If Keymin ≤ Pixel ≤ Keymax, the color is regarded as the colorkey. 1: If Pixel ≤ Keymin or Pixel ≥ Keymax, the color is regarded as the colorkey.
[14]	RW	key_en	Colorkey enable. 0: disabled 1: enabled
[13]	-	reserved	Reserved.
[12]	RW	palpha_en	Pixel alpha enable. 0: disabled 1: enabled

[11:9]	RW	reserved	Reserved.
[8]	RW	palpha_range	Range of the pixel alpha. 0: 0–128 1: 0–255
[7:0]	RW	galpha	Global blending alpha value. This value ranges from 0 to 127. The value 0 indicates full transparent and the value 127 indicates opaque.

G1CKEYMAX

G1CKEYMAX is the maximum colorkey value register of the G1 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0514	G1CKEYMAX	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	va0								keyr_max								keyg_max								keyb_max							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RW	va0	Alpha0 value. When the data format is alpha RGB1555 and the value of the MSB is 0, the alpha value is replaced with the value of alpha0.
[23:16]	RW	keyr_max	Maximum value of component R of the colorkey.
[15:8]	RW	keyg_max	Maximum value of component G of the colorkey.
[7:0]	RW	keyb_max	Maximum value of component B of the colorkey.

G1CKEYMIN

G1CKEYMIN is the minimum colorkey value register of the G1 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0518	G1CKEYMIN	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	va1								keyr_min								keyg_min								keyb_min							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RW	va1	Alpha1 value. When the data format is alpha RGB1555 and the value of the MSB is 0, the alpha value is replaced with the value of alpha1.
[23:16]	RW	keyr_min	Minimum value of component R of the colorkey.
[15:8]	RW	keyg_min	Minimum value of component G of the colorkey.
[7:0]	RW	keyb_min	Minimum value of component B of the colorkey.

G1IRESO

G1IRESO is the input resolution register of the G1 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x051C	G1IRESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	ih	Height of an input image at a graphics layer, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of an input image at a graphics layer must be an even number. There is no such restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	iw	Width of an input image at a graphics layer, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of an input image at a graphics layer must be an even number.

G1ORESO

G1ORESO is the output resolution register of the G1 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0520	G1ORESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	oh	Height of an output image at a graphics layer, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of an output image at a graphics layer must be an even number. There is no such restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	ow	Width of an output image at a graphics layer, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of an output image at a graphics layer must be an even number.

G1SFPOS

G1SFPOS is the source bitmap start position register of the G1 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0524	G1SFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	src_yfpos	Y coordinate of the source start coordinates. The value 0 indicates the first row. In interlaced output mode, the configured value must be an even number.

[15:12]	-	reserved	Reserved.
[11:0]	RW	src_xfpos	X coordinate of the source start coordinates. The value 0 indicates the first pixel in the first row.

G1DFPOS

G1DFPOS is the display window start position register of the G1 channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0528	G1DFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	yfpos	Column start coordinates.
[15:12]	-	reserved	Reserved.
[11:0]	RW	xfpos	Row start coordinates.

G1DLPOS

G1DLPOS is the display window end position register of the G1 channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x052C	G1DLPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	ylpos	Column end coordinates.
[15:12]	-	reserved	Reserved.
[11:0]	RW	xlpos	Row end coordinates.

G2CTRL

G2CTRL is the G2 channel control register. It is a non-instant register. It is used to configure the information about graphics layers.

Offset Address	Register Name	Total Reset Value
0x0600	G2CTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	g2_en	csc_en	csc_mo	p r e	resereved																		bitext		ifmt							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	g2_en	Surface enable. 0: disabled 1: enabled
[30]	RW	csc_en	Color space conversion enable. 0: disabled 1: enabled
[29]	RW	csc_mode	Color space conversion standard. 0: BT.601 1: BT.709
[28]	RW	pre_en	Data format premultiplied enable. 0: disabled 1: enabled
[27:10]	-	reserved	Reserved.
[9:8]	RW	bitext	Bit extend mode of the surface input bitmap. 0X: extend 0s to lower bits 10: extend the value of the MSB to a lower bit 11: extend the values of upper bits to lower bits
[7:0]	RW	ifmt	Input data format. 0x49: aRGB1555 Others: reserved

G2UPD

G2UPD is the G2 channel update enable register.

Offset Address	Register Name	Total Reset Value
0x0604	G2UPD	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	regup	Register update bit of surfaces. After the registers at graphics layer 2 are configured, the registers are updated when the value 1 is written to this bit. After updates, this bit is cleared automatically by the hardware.

G2ADDR

G2ADDR is the frame address register of the G2 channel.

Offset Address	Register Name	Total Reset Value
0x0608	G2ADDR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	g2addr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RW	g2addr	Buffer address for storing surface frames.

G2STRIDE

G2STRIDE is the frame stride register of the G2 channel.

Offset Address	Register Name	Total Reset Value
0x060C	G2STRIDE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																g2stride															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RW	g2stride	Row stride of the frame buffer.

G2CBMPARA

G2CBMPARA is the blending parameter register of the G2 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0610	G2CBMPARA	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																key_mode	key_en	reserved	palpha_en	reserved			palpha_range	galpha							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15]	RW	key_mode	Colorkey mode. 0: If Keymin ≤ Pixel ≤ Keymax, the color is regarded as the colorkey. 1: If Pixel ≤ Keymin or Pixel ≥ Keymax, the color is regarded as the colorkey.
[14]	RW	key_en	Colorkey enable. 0: disabled 1: enabled
[13]	-	reserved	Reserved.
[12]	RW	palpha_en	Pixel alpha enable. 0: disabled 1: enabled

[11:9]	-	reserved	Reserved.
[8]	RW	palpha_range	Range of the pixel alpha. 0: 0–128 1: 0–255
[7:0]	RW	galpha	Global blending alpha value. This value ranges from 0 to 127. The value 0 indicates full transparent and the value 127 indicates opaque.

G2CKEYMAX

G2CKEYMAX is the maximum colorkey value register of the G2 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0614	G2CKEYMAX	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	va0								keyr_max								keyg_max								keyb_max							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RW	va0	Alpha0 value. When the data format is alpha RGB1555 and the value of the MSB is 0, the alpha value is replaced with the value of alpha0.
[23:16]	RW	keyr_max	Maximum value of component R of the colorkey.
[15:8]	RW	keyg_max	Maximum value of component G of the colorkey.
[7:0]	RW	keyb_max	Maximum value of component B of the colorkey.

G2CKEYMIN

G2CKEYMIN is the minimum colorkey value register of the G2 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0618	G2CKEYMIN	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	va1		keyr_min		keyg_min		keyb_min	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:24]	RW	va1	Alpha1 value. When the data format is alpha RGB1555 and the value of the MSB is 0, the alpha value is replaced with the value of alpha1.
[23:16]	RW	keyr_min	Minimum value of component R of the colorkey.
[15:8]	RW	keyg_min	Minimum value of component G of the colorkey.
[7:0]	RW	keyb_min	Minimum value of component B of the colorkey.

G2IRESO

G2IRESO is the input resolution register of the G2 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x061C	G2IRESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	ih	Height of an input image at a graphics layer, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of an input image at a graphics layer must be an even number. There is no such restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	iw	Width of an input image at a graphics layer, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of an input image at a graphics layer must be an even number.

G2ORESO

G2ORESO is the output resolution register of the G2 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0620	G2ORESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	oh	Height of an output image at a graphics layer, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of an output image at a graphics layer must be an even number. There is no such restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	ow	Width of an output image at a graphics layer, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of an output image at a graphics layer must be an even number.

G2SFPOS

G2SFPOS is the source bitmap start position register of the G2 channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0624	G2SFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	src_yfpos	Y coordinate of the source start coordinates. The value 0 indicates the first row. In interlaced output mode, the configured value must be an even number.

[15:12]	-	reserved	Reserved.
[11:0]	RW	src_xfpos	X coordinate of the source start coordinates. The value 0 indicates the first pixel in the first row.

G2DFPOS

G2DFPOS is the display window start position register of the G2 channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0628	G2DFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	yfpos	Column start coordinates.
[15:12]	-	reserved	Reserved.
[11:0]	RW	xfpos	Row start coordinates.

G2DLPOS

G2DLPOS is the display window end position register of the G2 channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x062C	G2DLPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	ylpos	Column end coordinates.
[15:12]	-	reserved	Reserved.
[11:0]	RW	xlpos	Row end coordinates.

HCCTRL

HCCTRL is the HC channel control register. It is a non-instant register. It is used to configure the information about the HC layer.

Offset Address	Register Name	Total Reset Value
0x0800	HCCTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	hc_en	csc_en	csc_mode	reserved																			bitext		ifmt							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	hc_en	Surface enable. 0: disabled 1: enabled
[30]	RW	csc_en	Color space conversion enable. 0: disabled 1: enabled
[29]	RW	csc_mode	Color space conversion standard. 0: BT.601 1: BT.709
[28:10]	-	reserved	Reserved.
[9:8]	RW	bitext	Bit extend mode of the surface input bitmap. 0X: extend 0s to lower bits 10: extend the value of the MSB to a lower bit 11: extend the values of upper bits to lower bits
[7:0]	RW	ifmt	Input data format. 0x49: aRGB1555 Others: reserved

HCUPD

HCUPD is the HC channel update enable register.

Offset Address	Register Name	Total Reset Value
0x0804	HCUPD	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	regup	Register update bit of surfaces. After the registers at the HC layer are configured, the registers are updated when the value 1 is written to this bit. After updates, this bit is cleared automatically by the hardware.

HCADDR

HCADDR is the frame address register of the HC channel.

Offset Address	Register Name	Total Reset Value
0x0808	HCADDR	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	hcaddr	Buffer address for storing surface frames.

HCSTRIDE

HCSTRIDE is the frame stride address register of the HC channel.

Offset Address	Register Name	Total Reset Value
0x080C	HCSTRIDE	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RW	hcstride	Row stride of the frame buffer.

HCCBMPARA

HCCBMPARA is the blending parameter register of the HC channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0810	HCCBMPARA	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																key_mode	key_en	reserved	palpha_en	reserved			palpha_range	galpha								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15]	RW	key_mode	Colorkey mode. 0: If Keymin ≤ Pixel ≤ Keymax, the color is regarded as the colorkey. 1: If Pixel ≤ Keymin or Pixel ≥ Keymax, the color is regarded as the colorkey.
[14]	RW	key_en	Colorkey enable. 0: disabled 1: enabled
[13]	-	reserved	Reserved.
[12]	RW	palpha_en	Pixel alpha enable. 0: disabled 1: enabled

[11:9]	-	reserved	Reserved.
[8]	RW	palpha_range	Range of the pixel alpha. 0: 0–128 1: 0–255
[7:0]	RW	galpha	Global blending alpha value. This value ranges from 0 to 127. The value 0 indicates full transparent and the value 127 indicates opaque.

HCCKEYMAX

HCCKEYMAX is the maximum colorkey value register of the HC channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0814	HCCKEYMAX	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	va0		keyr_max		keyg_max		keyb_max	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:24]	RW	va0	Alpha0 value. When the data format is alpha RGB1555 and the value of the MSB is 0, the alpha value is replaced with the value of alpha0.
[23:16]	RW	keyr_max	Maximum value of component R of the colorkey.
[15:8]	RW	keyg_max	Maximum value of component G of the colorkey.
[7:0]	RW	keyb_max	Maximum value of component B of the colorkey.

HCCKEYMIN

HCCKEYMIN is the maximum colorkey value register of the HC channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0818	HCCKEYMIN	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	va1		keyr_min		keyg_min		keyb_min	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:24]	RW	va1	Alpha1 value. When the data format is alpha RGB1555 and the value of the MSB is 0, the alpha value is replaced with the value of alpha1.
[23:16]	RW	keyr_min	Minimum value of component R of the colorkey.
[15:8]	RW	keyg_min	Minimum value of component G of the colorkey.
[7:0]	RW	keyb_min	Minimum value of component B of the colorkey.

HCIRESO

HCIRESO is the input resolution register of the HC channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x081C	HCIRESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	ih	Height of an input image of the HC layer, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of the HC layer must be an even number. There is no such restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	iw	Width of an input image of the HC layer, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of the HC layer must be an even number.

HCORESO

HCORESO is the input resolution register of the HC channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0820	HCORESO	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	oh	Height of an output image of the HC layer, in the unit of row. The configured value is equal to the actual height minus 1. Note: In interlaced output mode, the actual height of the HC layer must be an even number. There is no such restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	ow	Width of an output image of the HC layer, in the unit of pixel. The configured value is equal to the actual width minus 1. Note: The actual width of the HC layer must be an even number.

HCSFPOS

HCSFPOS is the source bitmap start position register of the HC channel. This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0824	HCSFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	src_yfpos	Y coordinate of the source start coordinates. The value 0 indicates the first row. In interlaced output mode, the configured value must be an even number.
[15:12]	-	reserved	Reserved.

[11:0]	RW	src_xfpos	X coordinate of the source start coordinates. The value 0 indicates the first pixel in the first row.

HCFPOS

HCFPOS is the display window start position register of the HC channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x0828	HCFPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	disp_yfpos	Column start coordinates.
[15:12]	-	reserved	Reserved.
[11:0]	RW	disp_xfpos	Row start coordinates.

HCDLPOS

HCDLPOS is the display window end position register of the HC channel (in pixels). This register is a non-instant register.

Offset Address	Register Name	Total Reset Value
0x082C	HCDLPOS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:16]	RW	disp_ylpos	Column end coordinates.
[15:12]	-	reserved	Reserved.
[11:0]	RW	disp_xlpos	Row end coordinates.

CBMBKG1

CBMBKG1 is the blending background color register of mixer 1. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0B00	CBMBKG1	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								cbm_bkgcr1								cbm_bkgcb1								cbm_bkgy1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	-	reserved	Reserved.
[23:16]	RW	cbm_bkgcr1	Blending background color of mixer 1, component Y.
[15:8]	RW	cbm_bkgcb1	Blending background color of mixer 1, component Cb.
[7:0]	RW	cbm_bkgy1	Blending background color of mixer 1, component Cr.

CBMBKG2

CBMBKG2 is the blending background color register of mixer 2. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0B04	CBMBKG2	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								cbm_bkgcr2								cbm_bkgcb2								cbm_bkgy2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	-	reserved	Reserved.
[23:16]	RW	cbm_bkgcr2	Blending background color of mixer 2, component Y.
[15:8]	RW	cbm_bkgcb2	Blending background color of mixer 2, component Cb.
[7:0]	RW	cbm_bkgy2	Blending background color of mixer 2, component Cr.

CBCFG

CBCFG is the color bar configuration register. It is an instant register.

Offset Address	Register Name	Total Reset Value
0x0B0C	CBCFG	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved		sur_attr5	sur_attr4	sur_attr3	sur_attr2	sur_attr1	sur_attr0	reserved								mixer3_prio	mixer_prio4			mixer_prio3				mixer_prio2				mixer_prio1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29]	RW	sur_attr5	GDC_HC link. 0: mixer 1 1: mixer 2
[28]	RW	sur_attr4	GDC_G2 link. 0: mixer 1 1: mixer 2
[27]	RW	sur_attr3	GDC_G1 link. 0: mixer 1 1: mixer 2
[26]	RW	sur_attr2	GDC_G0 link. 0: mixer 1 1: reserved
[25]	RW	sur_attr1	VDC_AD link. 0: mixer 1 1: mixer 2
[24]	RW	sur_attr0	VDC_HD link. 0: mixer 1 1: reserved
[23:16]	-	reserved	Reserved.
[15]	RW	mixer3_prio	Blending layer priority of mixer 3. 0: The priority of G3 is higher than that of VSD. 1: The priority of VSD is higher than that of G3.

[14:12]	RW	mixer_prio4	Blended layer priority configuration of mixer 1 and mixer 2. It indicates the drive layer of priority 4 (the highest priority). 000: no drive layer 001: vdc_hd 010: vdc_ad 011: gdc_g0 100: gdc_g1 101: gdc_g2 Others: reserved
[11:9]	RW	mixer_prio3	Blended layer priority configuration of mixer 1 and mixer 2. It indicates the drive layer of priority 3. 000: no drive layer 001: vdc_hd 010: vdc_ad 011: gdc_g0 100: gdc_g1 101: gdc_g2 Others: reserved
[8:6]	RW	mixer_prio2	Blended layer priority configuration of mixer 1 and mixer 2. It indicates the drive layer of priority 2. 000: no drive layer 001: vdc_hd 010: vdc_ad 011: gdc_g0 100: gdc_g1 101: gdc_g2 Others: reserved
[5:3]	RW	mixer_prio1	Blended layer priority configuration of mixer 1 and mixer 2. It indicates the drive layer of priority 1. 000: no drive layer 001: vdc_hd 010: vdc_ad 011: gdc_g0 100: gdc_g1 101: gdc_g2 Others: reserved

[2:0]	RW	mixer_prio0	Blended layer priority configuration of mixer 1 and mixer 2. It indicates the drive layer of priority 0 (the lowest priority). 000: no drive layer 001: vdc_hd 010: vdc_ad 011: gdc_g0 100: gdc_g1 101: gdc_g2 Others: reserved

DHDCTRL

DHDCTRL is the HD display channel control register. It is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C00	DHDCTRL	0x0000_00EC

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	intf_en	slave_mo de	cbar_sel		reserved											clipen	cscen	gmmen	gmmmod e	reserved		idv	ihs	ivs	iop	synm	intfb		intfdm			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	0

Bits	Access	Name	Description
[31]	RW	intf_en	Display interface enable. 0: disabled 1: enabled Note: Data is output through the HD interface only when this interface is enabled.
[30]	RW	slave_mode	Slave mode enable of the display interface. 0: disabled (master mode) 1: enabled (driven by the input BT.1120 capture timing)
[29:28]	RW	cbar_sel	Color space control for the color bar. 00 and 01: RGB space 10: YUV space 11: RGB space
[27:17]	-	reserved	Reserved.
[16]	RW	clipen	Clip bit output enable. 0: disabled 1: enabled

[15]	RW	cscen	Color space conversion enable. 0: disabled 1: enabled
[14]	RW	gmmen	Output gamma correction enable. 0: disabled 1: enabled
[13]	RW	gmmmode	Output gamma correction mode. 0: The gamma table is generated by the hardware. 1: The gamma table is configured by the software.
[12:11]	-	reserved	Reserved.
[10]	RW	idv	Reverse phase output enable of the data valid signal. 0: disabled 1: enabled
[9]	RW	ihs	Reverse phase output enable of the horizontal sync pulse. 0: disabled 1: enabled
[8]	RW	ivs	Reverse phase output enable of the vertical sync pulse. 0: disabled 1: enabled
[7]	RW	iop	Interlaced or progressive display. 0: interlaced display 1: progressive display
[6]	RW	synm	Synchronization mode. 0: timing label mode (such as BT.656) 1: sync signal mode (such as LCD display)
[5:4]	RW	intfb	Bit width mode of the output interface. 00: single-component mode (each clock outputs one component) 01: 2-component mode (each clock outputs two components) 10: 3-component mode (each clock outputs three components) 11: reserved
[3:0]	RW	intfdm	Interface data format. 0x0: YCbCr4:2:2 0xC: RGB888/YCbCr4:4:4 output Others: reserved

DHDVSYNC

DHDVSYNC is the vertical sync timing register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C04	DHDVSYNC	0x0010_A257

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:20]	RW	vfb	In interlaced output mode, it indicates the top vertical blanking front porch. In progressive output mode, it indicates the vertical blanking front porch.
[19:12]	RW	vbb	In interlaced output mode, it indicates the top vertical blanking back porch. In progressive output mode, it indicates the vertical blanking back porch plus the vertical pulse width.
[11:0]	RW	vact	In interlaced output mode, it indicates the height of an active image in a top field. In progressive output mode, it indicates the height of an active image in a frame. The configured value is equal to the actual value minus 1.

DHDHSYNC1

DHDHSYNC1 is the horizontal sync timing register 1 of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C08	DHDHSYNC1	0x00D7_031F

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

Bits	Access	Name	Description
[31:16]	RW	hbb	Horizontal blanking back porch, in the unit of pixel.
[15:0]	RW	hact	Number of horizontal pixels in the active area.

DHDHSYNC2

DHDHSYNC2 is the horizontal sync timing register 2 of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C0C	DHDHSYNC2	0x0000_0027

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

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RW	hfb	Horizontal blanking front porch, in the unit of pixel.

DHDVPLUS

DHDVPLUS is the vertical sync plus timing register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C10	DHDVPLUS	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:20]	RW	bvfb	In interlaced output mode, it indicates the bottom vertical blanking front porch.
[19:12]	RW	bvbb	In interlaced output mode, it indicates the bottom vertical blanking back porch and the vertical pulse width.
[11:0]	RW	bvact	In interlaced output mode, it indicates the height of an active image in a bottom field. The configured value is equal to the actual value minus 1.

DHDPWR

DHDPWR is the sync pulse width register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C14	DHDPWR	0x0003_007F

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

Bits	Access	Name	Description
[31:24]	-	reserved	Reserved.
[23:16]	RW	vpw	The configured value is equal to the vertical pulse width minus 1.
[15:0]	RW	hpw	The configured value is equal to the horizontal pulse width minus 1.

DHDFIFOTHD

DHDFIFOTHD is the async FIFO threshold configuration register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C18	DHDFIFOTHD	0x0000_0000

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

Bits	Access	Name	Description
[31:12]	-	reserved	Reserved.
[11:0]	RW	aalmthd	Data alarm threshold. When the data amount in the async FIFO is below the threshold, an alarm is raised.

DHDVTTHD

DHDVTTHD is the vertical timing threshold register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C1C	DHDVTTHD	0x0000_0001

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

Bits	Access	Name	Description
[31:17]	-	reserved	Reserved.
[16]	RW	thd_mode	Threshold interrupt generation mode. 0: frame mode. The threshold count is in the unit of frame. 1: field mode. The threshold count is in the unit of field during interlace display.
[15:13]	-	reserved	Reserved.
[12:0]	RW	vtmgthd	Vertical timing threshold. When the vertical timing counter reaches this threshold, an HD channel vertical timing interrupt is triggered.

DHDCSCIDC

DHDCSCIDC is the input DC component for color space conversion register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C20	DHDCSCIDC	0x07C3_0180

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscidc2												cscidc1								cscidc0							
Reset	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:27]	-	reserved	Reserved.
[26:18]	RW	cscidc2	DC parameter of input component 2. The parameter value is expressed as a complementary code and the MSB is the sign bit.
[17:9]	RW	cscidc1	DC parameter of input component 1. The parameter value is expressed as a complementary code and the MSB is the sign bit.
[8:0]	RW	cscidc0	DC parameter of input component 0. The parameter value is expressed as a complementary code and the MSB is the sign bit.

DHDCSCODC

DHDCSCODC is the output DC component for color space conversion register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C24	DHDCSCODC	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	cscodc2		cscodc1		cscodc0		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:27]	-	reserved	Reserved.
[26:18]	RW	cscodc2	DC parameter of output component 2. The parameter value is expressed as a complementary code and the MSB is the sign bit.
[17:9]	RW	cscodc1	DC parameter of output component 1. The parameter value is expressed as a complementary code and the MSB is the sign bit.
[8:0]	RW	cscodc0	DC parameter of output component 0. The parameter value is expressed as a complementary code and the MSB is the sign bit.

DHDCSCP0

DHDCSCP0 is the color space conversion parameter 0 register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C28	DHDCSCP0	0x0000_012A

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	cscp01			reserved	cscp00		
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1	0 0 1 0	1 0 1 0

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	cscp01	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.
[15:13]	-	reserved	Reserved.
[12:0]	RW	cscp00	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DHDCSCP1

DHDCSCP1 is the color space conversion parameter 1 register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C2C	DHDCSCP1	0x012A_01CB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscp10												reserved				cscp02											
Reset	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	cscp10	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.
[15:13]	-	reserved	Reserved.
[12:0]	RW	cscp02	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DHDCSCP2

DHDCSCP2 is the color space conversion parameter 2 register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C30	DHDCSCP2	0x1F77_1FC9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscp12												reserved				cscp11											
Reset	0	0	0	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	1

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	cscp12	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.
[15:13]	-	reserved	Reserved.
[12:0]	RW	cscp11	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DHDCSCP3

DHDCSCP3 is the color space conversion parameter 3 register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C34	DHDCSCP3	0x021D_012A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscp21												reserved				cscp20											
Reset	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	cscp21	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.
[15:13]	-	reserved	Reserved.
[12:0]	RW	cscp20	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DHDCSCP4

DHDCSCP4 is the color space conversion parameter 4 register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C38	DHDCSCP4	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																				cscp22											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:13]	-	reserved	Reserved.
[12:0]	RW	cscp22	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DHDCLIPL

DHDCLIPL is the minimum threshold clip bit processing register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C3C	DHDCLIPL	0x0401_0040

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				clipcl2								clipcl1								clipcl0											
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:20]	RW	clipcl2	Minimum threshold Y/R of component 2, unsigned integer.
[19:10]	RW	clipcl1	Minimum threshold Cb/G of component 1, unsigned integer.
[9:0]	RW	clipcl0	Minimum threshold Cr/B of component 0, unsigned integer.

DHDCLIPH

DHDCLIPH is the maximum threshold clip bit processing register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C40	DHDCLIPH	0x3ACF_03C0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				clipch2								clipch1								clipch0											
Reset	0	0	1	1	1	0	1	0	1	1	1	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:20]	RW	clipch2	Maximum threshold Y/R of component 2, unsigned integer.
[19:10]	RW	clipch1	Maximum threshold Cb/G of component 1, unsigned integer.
[9:0]	RW	clipch0	Maximum threshold Cr/B of component 0, unsigned integer.

DHDGMMTHD1

DHDGMMTHD1 is the gamma operation threshold 1 register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C44	DHDGMMTHD1	0x0000_0000

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

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:20]	RW	thd_med_low	Threshold with med_low luminance.
[19:10]	RW	thd_high	Threshold with high luminance.
[9:0]	RW	thd_low	Threshold with low luminance.

DHDGMMTHD2

DHDGMMTHD2 is the gamma operation threshold 2 register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C48	DHDGMMTHD2	0x0000_0000

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

Bits	Access	Name	Description
[31:18]	-	reserved	Reserved.
[17:10]	RW	gmm_multiple	Gamma operation multiplier.
[9:0]	RW	thd_med_high	Threshold with med_high luminance.

DHDGMMLOWt

DHDGMMLOWt is the low-luminance gamma lookup table register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C50+t1×0x4	DHDGMMLOWt	0x0000_0000

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

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:0]	RW	table_datat	Gamma lookup table data t1.

DHDGMMMEDt

DHDGMMMEDt is the medium-luminance gamma lookup table register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C60+t2×0x4	DHDGMMMEDt	0x0000_0000

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

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:0]	RW	table_datat	Gamma lookup table data t2.

DHDGMMHIGHt

DHDGMMHIGHt is the high-luminance gamma lookup table register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C70+t3×0x4	DHDGMMHIGHt	0x0000_0000

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

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:0]	RW	table_datat	Gamma lookup table data t3.

DHDGMMMLt

DHDGMMMLt is the middle_low-luminance gamma lookup table register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0C80+t4×0x4	DHDGMMMLt	0x0000_0000

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

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:0]	RW	table_datat	Gamma lookup table data t4.

DHDGMMMHt

DHDGMMMHt is the middle_high-luminance gamma lookup table register of the HD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0CA0+t5×0x4	DHDGMMMHt	0x0000_0000

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

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:0]	RW	table_datat	Gamma lookup table data t5.

DHDGMM3LOW

DHDGMM3LOW is the low-threshold luminance statistics register of three gamma areas of the HD display channel.

Offset Address	Register Name	Total Reset Value
0x0CB0	DHDGMM3LOW	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												cnt3_low																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:0]	RO	cnt3_low	Low-threshold luminance statistics on three areas.

DHDGMM3MED

DHDGMM3MED is the medium-threshold luminance statistics register of three gamma areas of the HD display channel.

Offset Address	Register Name	Total Reset Value
0x0CB4	DHDGMM3MED	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												cnt3_med																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:0]	RO	cnt3_med	Medium-threshold luminance statistics on three areas.

DHDGMM3HIGH

DHDGMM3HIGH is the high-threshold luminance statistics register of three gamma areas of the HD display channel.

Offset Address	Register Name	Total Reset Value
0x0CB8	DHDGMM3HIGH	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												cnt3_high																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:0]	RO	cnt3_high	High-threshold luminance statistics on three areas.

DHDGMM8MLOW

DHDGMM8MLOW is the low-threshold luminance statistics register after the middle gamma area of the HD display channel is divided into eight segments.

Offset Address	Register Name	Total Reset Value
0x0CC0	DHDGMM8MLOW	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												cnt8_med_low																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.

[20:0]	RO	cnt8_med_low	Low-threshold luminance statistics on the middle area.

DHDGMM8MHIGH

DHDGMM8MHIGH is the high-threshold luminance statistics register after the middle gamma area of the HD display channel is divided into eight segments.

Offset Address	Register Name	Total Reset Value
0x0CC4	DHDGMM8MHIGH	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												cnt8_med_high																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:21]	-	reserved	Reserved.
[20:0]	RO	cnt8_med_high	High-threshold luminance statistics on the middle area.

DHDSTATE

DHDSTATE is the HD display channel status register.

Offset Address	Register Name	Total Reset Value
0x0CF0	DHDSTATE	0x0000_0006

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

Bits	Access	Name	Description
[31:3]	-	reserved	Reserved.
[2]	RO	bottom_field	DHD top/bottom field display flag. 0: top field 1: bottom field.
[1]	RO	vblank	DHD blanking area display flag. 0: valid area 1: blanking area

[0]	RO	vback_blank	DHD back blanking area display flag. 0: non-back blanking area 1: blanking area

DSDCTRL

DSDCTRL is the SD display channel control register. It is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E00	DSDCTRL	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	intf_en	reserved	cbar_sel		reserved											clipen	cscen	reserved			idv	ihs	ivs	iop	synm	intfb	intfdm					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	intf_en	Display interface enable. 0: disabled 1: enabled Note: Data is output through the SD interface only when this interface is enabled.
[30]	-	reserved	Reserved.
[29:28]	RW	cbar_sel	Color space control for the color bar. 00 and 01: YUV space 10: YUV space 11: RGB space
[27:17]	-	reserved	Reserved.
[16]	RW	clipen	Clip bit output enable. 0: disabled 1: enabled
[15]	RW	cscen	Color space conversion enable. 0: disabled 1: enabled
[14:11]	-	reserved	Reserved.

[10]	RW	idv	Reverse phase output enable of the data valid signal. 0: disabled 1: enabled
[9]	RW	ihs	Reverse phase output enable of the horizontal sync pulse. 0: disabled 1: enabled
[8]	RW	ivs	Reverse phase output enable of the vertical sync pulse. 0: disabled 1: enabled
[7]	RW	iop	Interlaced or progressive display. 0: interlaced display 1: progressive display
[6]	RW	synm	Synchronization mode. 0: timing label mode (such as BT.656) 1: sync signal mode (such as LCD display)
[5:4]	RW	intfb	Bit width mode of the output interface. 00: single-component mode (each clock outputs one component) 01: 2-component mode (each clock outputs two components) 10: 3-component mode (each clock outputs three components) 11: reserved
[3:0]	RW	intfdm	Interface data format. 0x0: YCbCr4:2:2 0xC: RGB888/YCbCr4:4:4 output Others: reserved

DSDVSYNC

DSDVSYNC is the vertical sync timing register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E04	DSDVSYNC	0x0011_511F

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.

[27:20]	RW	vfb	In interlaced output mode, it indicates the top vertical blanking front porch. In progressive output mode, it indicates the vertical blanking front porch.
[19:12]	RW	vbb	In interlaced output mode, it indicates the top vertical blanking back porch. In progressive output mode, it indicates the vertical blanking back porch plus the vertical pulse width.
[11:0]	RW	vact	In interlaced output mode, it indicates the height of an active image in a top field. In progressive output mode, it indicates the height of an active image in a frame. The configured value is equal to the actual height minus 1.

DSDHSYNC1

DSDHSYNC1 is the horizontal sync timing register 1 of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E08	DSDHSYNC1	0x0107_02CF

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

Bits	Access	Name	Description
[31:16]	RW	hbb	Horizontal blanking back porch, in the unit of pixel.
[15:0]	RW	hact	Number of horizontal pixels in the active area.

DSDHSYNC2

DSDHSYNC2 is the horizontal sync timing register 2 of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E0C	DSDHSYNC2	0x0000_0017

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

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RW	hfb	Horizontal blanking front porch, in the unit of pixel.

DSDVPLUS

DSDVPLUS is the vertical sync plus timing register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E10	DSDVPLUS	0x0011_611F

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

Bits	Access	Name	Description
[31:28]	-	reserved	Reserved.
[27:20]	RW	bvfb	In interlaced output mode, it indicates the bottom vertical blanking front porch.
[19:12]	RW	bvbb	In interlaced output mode, it indicates the bottom vertical blanking back porch and the vertical pulse width.
[11:0]	RW	bvact	In interlaced output mode, it indicates the height of an active image in a bottom field. The configured value is equal to the actual height minus 1.

DSDPWR

DSDPWR is the sync pulse width register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E14	DSDPWR	0x0000_0000

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

Bits	Access	Name	Description
[31:24]	-	reserved	Reserved.
[23:16]	RW	vpw	The configured value is equal to the vertical pulse width minus 1.
[15:0]	RW	hpw	The configured value is equal to the horizontal pulse width minus 1.

DSDFIFOTHD

DSDFIFOTHD is the async FIFO threshold configuration register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E18	DSDFIFOTHD	0x0000_0000

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

Bits	Access	Name	Description
[31:12]	-	reserved	Reserved.
[11:0]	RW	aalmthd	Data alarm threshold. When the data amount in the async FIFO is below the threshold, an alarm is raised.

DSDVTTHD

DSDVTTHD is the vertical timing threshold register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E1C	DSDVTTHD	0x0000_0001

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

Bits	Access	Name	Description
[31:17]	-	reserved	Reserved.
[16]	RW	thd_mode	Threshold interrupt generation mode. 0: frame mode. The threshold count is in the unit of frame. 1: field mode. The threshold count is in the unit of field during interlace display.
[15:13]	-	reserved	Reserved.
[12:0]	RW	vtmgthd	Vertical timing threshold. When the vertical timing counter reaches this threshold, an SD channel vertical timing interrupt is triggered.

DSDCSCIDC

DSDCSCIDC is the input DC component for color space conversion register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E20	DSDCSCIDC	0x07C3_0180

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscidc2												cscidc1								cscidc0							
Reset	0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:27]	-	reserved	Reserved.
[26:18]	RW	cscidc2	DC parameter of input component 2. The MSB is the sign bit. The parameter value is expressed as a complementary code.
[17:9]	RW	cscidc1	DC parameter of input component 1. The MSB is the sign bit. The parameter value is expressed as a complementary code.
[8:0]	RW	cscidc0	DC parameter of input component 0. The MSB is the sign bit. The parameter value is expressed as a complementary code.

DSDCSCODC

DSDCSCODC is the output DC component for color space conversion register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E24	DSDCSCODC	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscodc2												cscodc1								cscodc0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:27]	-	reserved	Reserved.
[26:18]	RW	cscodc2	DC parameter of output component 2. The MSB is the sign bit. The parameter value is expressed as a complementary code.
[17:9]	RW	cscodc1	DC parameter of output component 1. The MSB is the sign bit. The parameter value is expressed as a complementary code.
[8:0]	RW	cscodc0	DC parameter of output component 0. The MSB is the sign bit. The parameter value is expressed as a complementary code.

DSDCSCP0

DSDCSCP0 is the color space conversion parameter 0 register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E28	DSDCSCP0	0x0000_012A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscp01												reserved				cscp00											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	cscp01	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.
[15:13]	-	reserved	Reserved.
[12:0]	RW	cscp00	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DSDCSCP1

DSDCSCP1 is the color space conversion parameter 1 register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E2C	DSDCSCP1	0x012A_01CB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscp10												reserved				cscp02											
Reset	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1	1

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	cscp10	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.
[15:13]	-	reserved	Reserved.
[12:0]	RW	cscp02	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DSDCSCP2

DSDCSCP2 is the color space conversion parameter 2 register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E30	DSDCSCP2	0x1F77_1FC9

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscp12												reserved				cscp11											
Reset	0	0	0	1	1	1	1	1	0	1	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	1	0	0	1

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	cscp12	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.
[15:13]	-	reserved	Reserved.
[12:0]	RW	cscp11	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DSDCSCP3

DSDCSCP3 is the color space conversion parameter 3 register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E34	DSDCSCP3	0x021D_012A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				cscp21												reserved				cscp20											
Reset	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0

Bits	Access	Name	Description
[31:29]	-	reserved	Reserved.
[28:16]	RW	cscp21	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.
[15:13]	-	reserved	Reserved.
[12:0]	RW	cscp20	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DSDCSCP4

DSDCSCP4 is the color space conversion parameter 4 register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E38	DSDCSCP4	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																				cscp22											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:13]	-	reserved	Reserved.
[12:0]	RW	cscp22	5.8 data format: 1-bit sign bit, 4-bit integer bit, and 8-bit decimal bit. The value is expressed as a complementary code.

DSDCLIPL

DSDCLIPL is the minimum threshold clip bit processing register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E3C	DSDCLIPL	0x0401_0040

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				clipcl2								clipcl1								clipcl0											
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:20]	RW	clipcl2	Minimum threshold Y/R of component 2, unsigned integer.
[19:10]	RW	clipcl1	Minimum threshold Cb/G of component 1, unsigned integer.
[9:0]	RW	clipcl0	Minimum threshold Cr/B of component 0, unsigned integer.

DSDCLIPH

DSDCLIPH is the maximum threshold clip bit processing register of the SD display channel. This register is an instant register.

Offset Address	Register Name	Total Reset Value
0x0E40	DSDCLIPH	0x3ACF_03C0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				clipch2								clipch1								clipch0											
Reset	0	0	1	1	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0

Bits	Access	Name	Description
[31:30]	-	reserved	Reserved.
[29:20]	RW	clipch2	Maximum threshold Y/R of component 2, unsigned integer.
[19:10]	RW	clipch1	Maximum threshold Cb/G of component 1, unsigned integer.
[9:0]	RW	clipch0	Maximum threshold Cr/B of component 0, unsigned integer.

DSDSTATE

DSDSTATE is the SD display channel status register.

Offset Address	Register Name	Total Reset Value
0x0EF0	DSDSTATE	0x0000_0006

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

Bits	Access	Name	Description
[31:3]	-	reserved	Reserved.
[2]	RW	bottom_field	DSD top/bottom field display flag. 0: top field 1: bottom field.
[1]	RW	vblank	DSD blanking area display flag. 0: valid area 1: blanking area
[0]	RW	vback_blank	DSD back blanking area display flag. 0: non-back blanking area 1: blanking area

VHDHLCOEF

VHDHLCOEF is the luminance horizontal scaling and filtering coefficient register of the VHD channel. There are 18 groups of registers and each group consists of four registers. The lower 4-bit addresses of the four registers are 0x0, 0x4, 0x8, and 0xC respectively. The register value indicates 8-order coefficients.

Offset Address	Register Name	Total Reset Value
0x1000–0x111C	VHDHLCOEF	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								hlcoefn2								reserved				hlcoefn1											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:26]	-	reserved	Reserved.

[25:16]	RW	hlcoefn2	Luminance horizontal scaling and filtering coefficient. Indicate the second order coefficients when bit[3:0] of the register address are 0x0. Indicate the fourth order coefficients when bit[3:0] of the register address are 0x4. Indicate the sixth order coefficients when bit[3:0] of the register address are 0x8. Indicate the eighth order coefficients when bit[3:0] of the register address are 0xC.
[15:10]	-	reserved	Reserved.
[9:0]	RW	hlcoefn1	Luminance horizontal scaling and filtering coefficient. Indicate the first order coefficients when bit[3:0] of the register address is 0x0. Indicate the third order coefficients when bit[3:0] of the register address are 0x4. Indicate the fifth order coefficients when bit[3:0] of the register address are 0x8. Indicate the seventh order coefficients when bit[3:0] of the register address are 0xC.

VHDHCCOEF

VHDHCCOEF is the chrominance horizontal scaling and filtering coefficient register of the VHD channel. There are 18 groups of registers and each group consists of two registers. The lower 4-bit addresses of the two registers are 0x0 and 0x4 or 0x8 and 0xC respectively. The register value indicate 4-order coefficients.

Offset Address	Register Name	Total Reset Value
0x1200–0x128C	VHDHCCOEF	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								hccoefn2								reserved								hccoefn1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:26]	-	reserved	Reserved.
[25:16]	RW	hccoefn2	Chrominance horizontal scaling and filtering coefficient. Indicate the second order coefficients when bit[3:0] of the register address are 0x0 or 0x8. Indicate the fourth order coefficients when bit[3:0] of the register address are 0x4 or 0xC.
[15:10]	-	reserved	Reserved.

[9:0]	RW	hccoefn1	Chrominance horizontal scaling and filtering coefficient. Indicate the first order coefficients when bit[3:0] of the register address are 0x0 or 0x8. Indicate the third order coefficients when bit[3:0] of the register address are 0x4 or 0xC.

VHDVLCOEF

VHDVLCOEF is the luminance vertical scaling and filtering coefficient register of the VHD channel. There are 18 groups of registers and each group consists of two registers. The lower 4-bit addresses of the two registers are 0x0 and 0x4 or 0x8 and 0xC respectively. The register value indicate 4-order coefficients.

Offset Address	Register Name	Total Reset Value
0x1300–0x138C	VHDVLCOEF	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	vlcoefn2			reserved		vlcoefn1	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:26]	-	reserved	Reserved.
[25:16]	RW	vlcoefn2	Luminance vertical scaling and filtering coefficient. Indicate the second order coefficients when bit[3:0] of the register address are 0x0 or 0x8. Indicate the fourth order coefficients when bit[3:0] of the register address are 0x4 or 0xC.
[15:10]	-	reserved	Reserved.
[9:0]	RW	vlcoefn1	Luminance vertical scaling and filtering coefficient. Indicate the first order coefficients when bit[3:0] of the register address are 0x0 or 0x8. Indicate the third order coefficients when bit[3:0] of the register address are 0x4 or 0xC.

VHDVCCOEF

VHDVCCOEF is the chrominance vertical scaling and filtering coefficient register of the VHD channel. There are 18 groups of registers and each group consists of two registers. The lower 4-bit addresses of the two registers are 0x0 and 0x4 or 0x8 and 0xC respectively. The register value indicate 4-order coefficients.

Offset Address	Register Name	Total Reset Value
0x1400–0x148C	VHDVCCOEF	0x0000_0000

Bit	31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved		vccoefn2		reserved		vccoefn1	
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31:26]	-	reserved	Reserved.
[25:16]	RW	vccoefn2	Luminance vertical scaling and filtering coefficient. Indicate the second order coefficients when bit[3:0] of the register address are 0x0 or 0x8. Indicate the fourth order coefficients when bit[3:0] of the register address are 0x4 or 0xC.
[15:10]	-	reserved	Reserved.
[9:0]	RW	vccoefn1	Luminance vertical scaling and filtering coefficient. Indicate the first order coefficients when bit[3:0] of the register address are 0x0 or 0x8. Indicate the third order coefficients when bit[3:0] of the register address are 0x4 or 0xC.

VHDMIMGSPOSp

VHDMIMGSPOSp is the start position and valid flag register of subimage p in the de-interlace partition of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x2300–0x237C	VHDMIMGSPOSp	0x0000_0000

Bit	31	30	29	28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Name	reserved	die_valid	die_mode	reserved	spos_y			reserved	spos_x		
Reset	0	0	0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bits	Access	Name	Description
[31]	-	reserved	Reserved.
[30]	RW	die_valid	De-interlace valid flag of subimage p. 0: invalid. 1: valid

[29]	RW	die_mode	De-interlace mode of subimage p. 0: The subimage p is processed based on the configuration of the VHDDIECTRL register 1: The subimage p is processed in 2-field median mode.
[28]	-	reserved	Reserved.
[27:16]	RW	spos_y	Vertical start coordinates, in the unit of row. Note: In interlaced output mode, the actual coordinates must be even numbers. There is no restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	spos_x	Horizontal start coordinates, in the unit of pixel. Note: The actual coordinates must be even numbers.

VHDMIMGFPOSp

VHDMIMGFPOSp is the end position register of sub image p in the de-interlace partition of the VHD channel.

Offset Address	Register Name	Total Reset Value
0x2380–0x23FC	VHDMIMGFPOSp	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	RW	reserved	Reserved.
[27:16]	RW	fpos_y	Vertical end coordinates, in the unit of row. Note: In interlaced output mode, the actual coordinates must be odd numbers. There is no restriction in progressive output mode.
[15:12]	-	reserved	Reserved.
[11:0]	RW	fpos_x	Horizontal end coordinates, in the unit of pixel. Note: The actual coordinates must be odd numbers.

DHDVBIPOS1

DHDVBIPOS1 is the position and start coordinate register of VBI information 1 of the DHD channel.

Offset Address	Register Name	Total Reset Value
0x2600	DHDVBISPOS1	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vbi_en	top_filed	vbb_region	reserved					spos_y								reserved				spos_x											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	vbi_en	VBI information 1 output enable. 0: disabled 1: enabled
[30]	RW	top_filed	Top/bottom flag bit in the VBI information. This bit is invalid in progressive display mode. 0: bottom field 1: top field
[29]	RW	Vbb_region	Front blanking flag bit in the VBI information. 0: VFB 1: VBB
[28:24]	-	reserved	Reserved.
[23:16]	RW	spos_y	Vertical start coordinates, in the unit of row. The value is the actual height minus 1.
[15:11]	-	reserved	Reserved.
[10:0]	RW	spos_x	Horizontal start coordinates, in the unit of clock. The value is the actual clock width minus 1.

DHDVBIPOS2

DHDVBIPOS2 is the position and start coordinate register of VBI information 2 of the DHD channel.

Offset Address	Register Name	Total Reset Value
0x2604	DHDVBISPOS2	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vbi_en	top_filed	vbb_region	reserved					spos_y								reserved				spos_x											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	vbi_en	VBI information 2 output enable. 0: disabled 1: enabled
[30]	RW	top_filed	Top/bottom flag bit in the VBI information. This bit is invalid in progressive display mode. 0: bottom field 1: top field
[29]	RW	vbb_region	Front blanking flag bit in the VBI information. 0: VFB 1: VBB
[28:24]	-	reserved	Reserved.
[23:16]	RW	spos_y	Vertical start coordinates, in the unit of row. The value is the actual height minus 1.
[15:11]	-	reserved	Reserved.
[10:0]	RW	spos_x	Horizontal start coordinates, in the unit of clock. The value is the actual clock width minus 1.

DHDVBIF1n

DHDVBIF1n is the content register of VBI information 1 of the DHD channel.

Offset Address	Register Name	Total Reset Value
0x2610–262C	DHDVBI1F0–DHDVBI1F7	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vbi1_fn3								vbi1_fn2								vbi1_fn1								vbi1_fn0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RW	Vbi1_fn3	Content n0 of VBI information 1.
[23:16]	RW	Vbi1_fn2	Content n1 of VBI information 1.
[15:8]	RW	Vbi1_fn1	Content n2 of VBI information 1.
[7:0]	RW	Vbi1_fn0	Content n3 of VBI information 1.

DHDVBIF2n

DHDVBIF2n is the content register of VBI information 2 of the DHD channel.

Offset Address	Register Name	Total Reset Value
0x2630–264C	DHDVBI2F0–DHDVBI2F7	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vbi2_fn3								vbi2_fn2								vbi2_fn1								vbi2_fn0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RW	vbi2_fn3	Content n0 of VBI information 2.
[23:16]	RW	vbi2_fn2	Content n1 of VBI information 2.
[15:8]	RW	vbi2_fn1	Content n2 of VBI information 2.
[7:0]	RW	vbi2_fn0	Content n3 of VBI information 2.

DSDVBIPOS1

DSDVBIPOS1 is the position and start coordinate register of VBI information 1 of the DSD channel.

Offset Address	Register Name	Total Reset Value
0x2700	DSDVBISPOS1	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vbi_en	top_filed	vbb_region	reserved					spos_y								reserved				spos_x											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	vbi_en	VBI information 1 output enable. 0: disabled 1: enabled
[30]	RW	top_filed	Top/bottom flag bit in the VBI information. This bit is invalid in progressive display mode. 0: bottom field 1: top field
[29]	RW	Vbb_region	Front blanking flag bit in the VBI information. 0: VFB 1: VBB
[28:24]	-	reserved	Reserved.
[23:16]	RW	spos_y	Vertical start coordinates, in the unit of row. The value is the actual height minus 1.
[15:11]	-	reserved	Reserved.
[10:0]	RW	spos_x	Horizontal start coordinates, in the unit of clock. The value is the actual clock width minus 1.

DSDVBIPOS2

DSDVBIPOS2 is the position and start coordinate register of VBI information 2 of the DSD channel.

Offset Address	Register Name	Total Reset Value
0x2704	DSDVBISPOS2	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vbi_en	top_filed	vbb_region	reserved					spos_y								reserved				spos_x											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	vbi_en	VBI information 2 output enable. 0: disabled 1: enabled
[30]	RW	top_filed	Top/bottom flag bit in the VBI information. This bit is invalid in progressive display mode. 0: bottom field 1: top field
[29]	RW	Vbb_region	Front blanking flag bit in the VBI information. 0: VFB 1: VBB
[28:24]	-	reserved	Reserved.
[23:16]	RW	spos_y	Vertical start coordinates, in the unit of row. The value is the actual height minus 1.
[15:11]	-	reserved	Reserved.
[10:0]	RW	spos_x	Horizontal start coordinates, in the unit of clock. The value is the actual clock width minus 1.

DSDVBIF1n

DSDVBIF1n is the content register of VBI information 1 of the DSD channel.

Offset Address	Register Name	Total Reset Value
0x2710–272c	DSDVBI1F0–DSDVBI1F7	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vbi_fn3								vbi_fn2								vbi_fn1								vbi_fn1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RW	vbi1_fn3	Content n0 of VBI information 1.
[23:16]	RW	vbi1_fn2	Content n1 of VBI information 1.
[15:8]	RW	vbi1_fn1	Content n2 of VBI information 1.
[7:0]	RW	vbi1_fn0	Content n3 of VBI information 1.

DSDVBIF2n

DSDVBIF2n is the content register of VBI information 2 of the DSD channel.

Offset Address	Register Name	Total Reset Value
0x2730–274C	DSDVBI2F0–DSDVBI2F7	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vbi2_fn3								vbi2_fn2								vbi2_fn1								vbi2_fn0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:24]	RW	vbi2_fn3	Content n0 of VBI information 2.
[23:16]	RW	vbi2_fn2	Content n1 of VBI information 2.
[15:8]	RW	vbi2_fn1	Content n2 of VBI information 2.
[7:0]	RW	vbi2_fn0	Content n3 of VBI information 2.

Contents

Figures

Tables

7 Audio Interface

7.1 Overview

The sonic input/output (SIO) interface is used to connect to the off-chip audio codec to play and record the music (voice). The Hi3515 provides two SIO interfaces: SIO0 and SIO1. SIO0 is used to input or output audios to implement talkback voice communication, whereas SIO1 is used to input the audios of multiple channels.

7.2 Features

The SIO interface supports both the pulse code modulation (PCM) interface and I^2S interfaces. The PCM interface is mainly used for voice channels, such as the VOIP phone; the I^2S interface is used to work with the audio codec to play and record music. The SIO interface also supports the directory memory access (DMA) operation.

7.2.1 Features of the PCM Interface

The PCM interface has the following features:

- Supports the master/slave mode.
- Transmits/receives the single-channel 8-/16-bit linear PCM code.
- Receives 2-/4-/8-/16-channel 8-/16-bit data.
- The bit clock and frame sync signal can be internally generated or externally supplied.
- The frame sync signals of the PCM interface are short pulse sync signals (referring to those sync signals with one clock cycle duration) only. The PCM interface can work in both standard mode and customized mode.
- Supports the function of enabling the transmit channel and the receive channel independently.
- Provides separate FIFOs for the receive channel and transmit channel. Each of them is 16-location deep.

7.2.2 Features of the I²S Interface

The I^2S interface has the following features:

- Supports the master/slave mode.

- Transmits or receives the left-/right-channel 16-/18-/20-/24-/32-bit data.
- Receives 2-/4-/8-/16-channel 8-/16-bit data.
- Supports the sampling rate ranging from 8 kHz to 192 kHz.
- Provides separate FIFOs for the I^2S transmit and receive channels and separate FIFOs for the left and right audio channels. Each of the FIFO is 16-location deep and the FIFO threshold is adjustable.
- Supports the function of enabling the transmit channel and the receive channel independently.
- For the 16-bit data transfer through the I^2S interface, the received left-channel and right-channel data can be merged into 32-bit wide and then stored in the receive FIFO (RX_FIFO). Additionally, the transmitted left-channel and right-channel data can be merged into 32-bit wide and then written into the transmit FIFO (TX_FIFO). Therefore, the buffer capacity of the FIFO is increased. The merging function is not supported when 2-/4-/8-/16-channel 8-/16-bit data is received.

7.3 Signal Description

The SIO module provides two SIO interfaces: SIO0 and SIO1. Table 7-1 describes the signals of the two SIO interfaces.

Table 7-1 SIO interface signals

Signal Name	Direction	Description	Corresponding Pin
SIO0_DI	I	Data input.	SIO0DI
SIO0_DO	O	Data output.	SIO0DO
SIO0_XFS	I/O	I^2S transmit audio channel select signal (when connected to the DAC interface) or PCM frame sync signal. The signal is multiplexed with the GPIO pins. For details about the multiplexing configuration, see "Pin Multiplexing" in section 7.5 .	SIO0XFS
SIO0_RFS	I/O	I^2S receive audio channel select signal (when connected to the ADC interface) or PCM frame sync signal.	SIO0RFS
SIO0_XCK	I/O	I^2S or PCM transmit bit stream clock. The signal is multiplexed with the GPIO pins. For details about the multiplexing configuration, see "Pin Multiplexing" in section 7.5 .	SIO0XCK
SIO0_RCK	I/O	I^2S or PCM receive bit stream clock.	SIO0RCK
SIO_MCLK	O	Main clock of the I^2S or PCM interface. It acts as the working clock of the codec. The signal is multiplexed with the GPIO pins. For details about the multiplexing configuration, see "Pin Multiplexing" in section 7.5 .	ACKOUT

Signal Name	Direction	Description	Corresponding Pin
SIO1_DI	I	Data input.	SIO1DI
SIO1_RFS	I/O	I^2S receive audio channel select signal (when connected to the ADC interface) or PCM frame sync signal.	SIO1RFS
SIO1_RCK	I/O	I^2S or PCM receive bit stream clock.	SIO1RCK

For certain audio codecs, the transmit audio channel select signal and the receive audio channel select signal are the same. In this case, when such codecs are connected, only SIO0_RFS needs to be connected. The internal XFS signal of SIO0 is obtained through the RFS by setting the system controller register SC_PERCTRL12[sio0_xfs] to 0b1. In this way, SIO0_XFS can be used as a GPIO pin. The bit clock can be configured through SC_PERCTRL12[sio0_xck].

7.4 Function Description

Typical Application

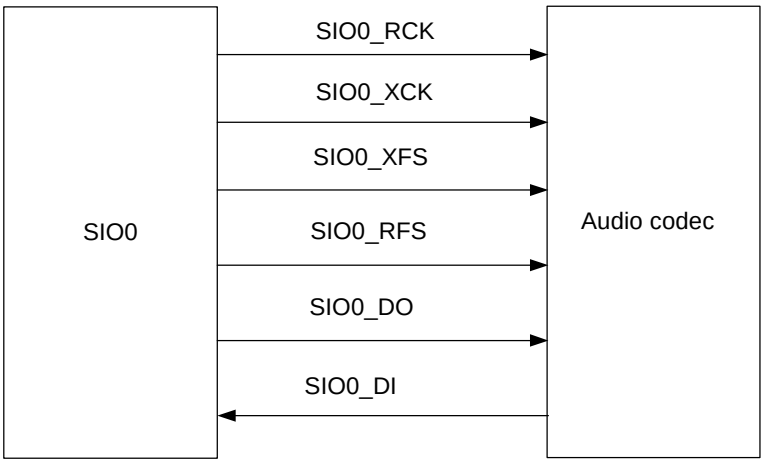
SIO0 is used to input or output audios to implement talkback voice communication. The following describes the typical connections of the I^2S interface.

CAUTION

In master mode, the working clock of the audio codec uses the main clock (SIO_MCLK) signal provided by the Hi3515 rather than the external crystal oscillator. Otherwise, sound distortion may occur.

Figure 7-1 shows the typical connections of the I^2S interface in master mode.

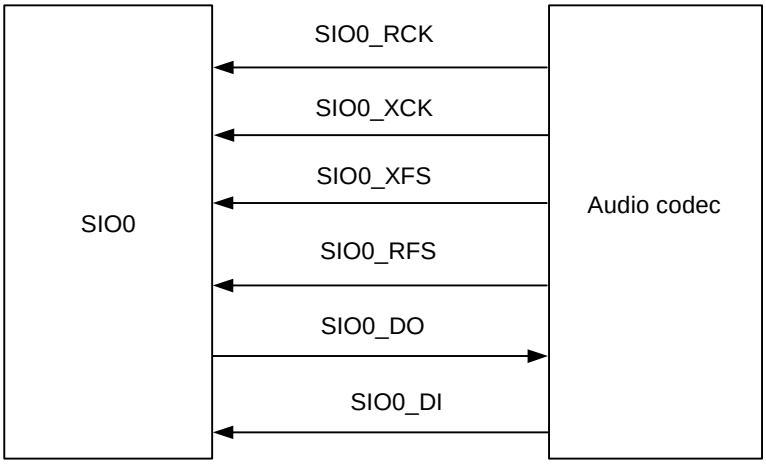
Figure 7-1 Connection diagram of the I²S interface in master mode



In master mode, the bit stream clock and audio channel select signal are sent to the audio codec by the SIO. In slave mode, the bit stream clock and audio channel select signal are sent to the SIO by the audio codec.

Figure 7-2 shows the typical connections of the I²S interface in slave mode.

Figure 7-2 Connections diagram of the I²S interface in slave mode



In slave mode, the main working clock of the audio codec can use both the main clock signal (SIO_MCLK) provided by the Hi3515 and the external crystal oscillator

Figure 7-3 shows the typical connections of the PCM interface when the SIO provides the clock and sync signals. The working clock of the audio codec uses the SIO_MCLK signal provided by the Hi3515 rather than the external crystal oscillator. Otherwise, the sound distortion may occur.

Figure 7-1 shows the connections of the PCM interface when the SIO provides the clock and sync signals in master mode. The PCM interface is connected in the same manner as the I²S interface.

Figure 7-2 shows the connections of the PCM interface when the audio codec provides the clock and sync signals in slave mode. The PCM interface is connected in the same manner as the I²S interface.

When the clock and sync signals are provided by the audio codec, the main work clock of the audio codec can use both the main clock (SIO_MCLK) signal provided by the Hi3515 and the external crystal oscillator.

SIO1 is used to record 8-/16-channel 16-bit audio data. Figure 7-3 shows the connections of the I²S/PCM interface used for audio recording in master mode. Figure 7-4 shows the connections of the I²S/PCM interface used for audio recording in slave mode.

Figure 7-3 Connections of the I²S/PCM interface used for audio recording in master mode

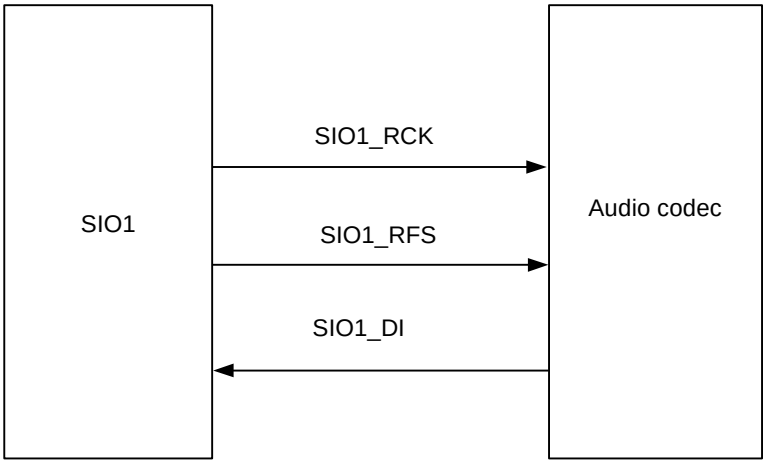
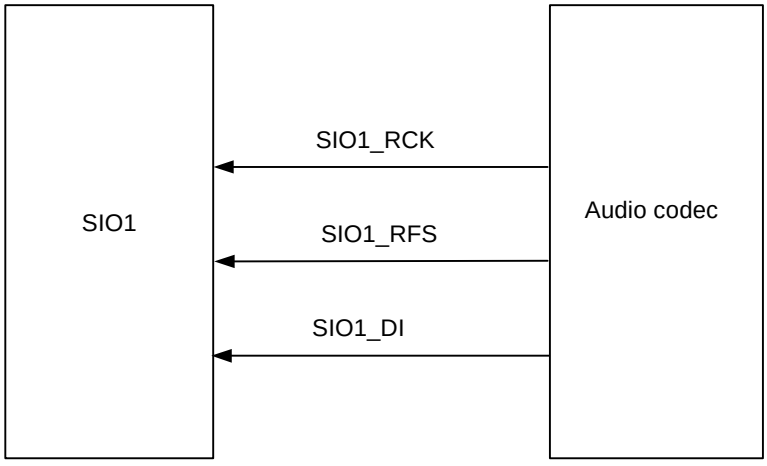


Figure 7-4 Connections of the I²S/PCM interface used for audio recording in slave mode



In audio recording, the function of receiving multi-channel audios is supported. That is, 8-/16-bit data of 2-/4-/8-/16-channel can be received.

Function Principle

An SIO interface receives the audio data from the internal bus, and then sends the audio data to the interconnected audio codec through the I^2S or PCM interface at a specified sampling rate. After that, the audio codec performs digital-to-analog (DA) conversion on the audio data and then plays the audio. At the same time, the SIO interface receives the audio data that has experienced the AD conversion performed by the audio codec through the I^2S and PCM interface, and stores the data into the internal FIFO. Then, the CPU takes the data and stores it. In this way, audio recording is implemented.

The data transferred through the I^2S interface consists of the right-channel data and the left-channel data, which are distinguished from the levels of the XFS (RFS) signal. See Figure 7-5. According to the related protocol, data is sampled on the rising edge of the XCK/RCK clock. The most significant bit (MSB) is valid in the next XFS/RFS clock cycle. The data is transferred in the sequence from the MSB to the least significant bit (LSB).

Figure 7-5 shows the timing diagram of the I^2S interface.

Figure 7-5 Timing diagram of the I2S interface

The data transferred through the PCM interface is single-channel data. XFS identifies the start of the data. The MSB is transmitted or received first and data is sampled on the falling edge. In standard timing mode, the MSB is valid in the next cycle after the high-level pulse of XFS; in customized timing mode, the MSB is aligned with the high-level pulse of XFS.

Figure 7-6 shows the timing diagram of the PCM interface in standard mode.

Figure 7-6 Timing diagram of the PCM interface in standard mode

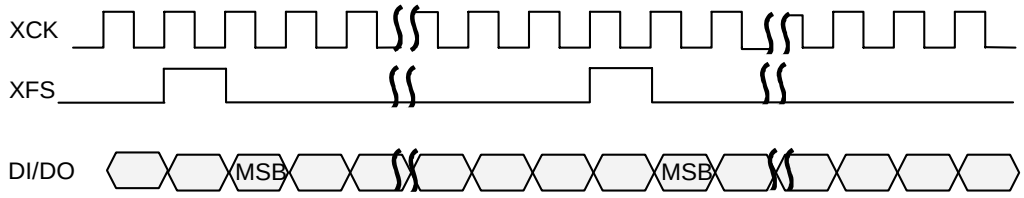
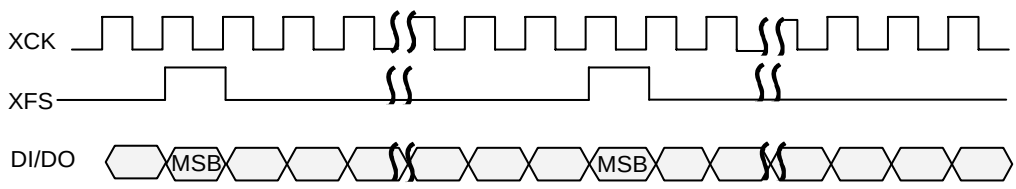


Figure 7-7 shows the timing diagram of the PCM interface in customized mode.

Figure 7-7 Timing diagram of the PCM interface in customized mode



When receiving 2-/4-/8-/16-channel 8-/16 bit data, the I²S interface stores the data into the left and right audio channels respectively. See Figure 7-8.

Figure 7-8 Receiving 2-/4-/8-/16-channel data through the I²S interface

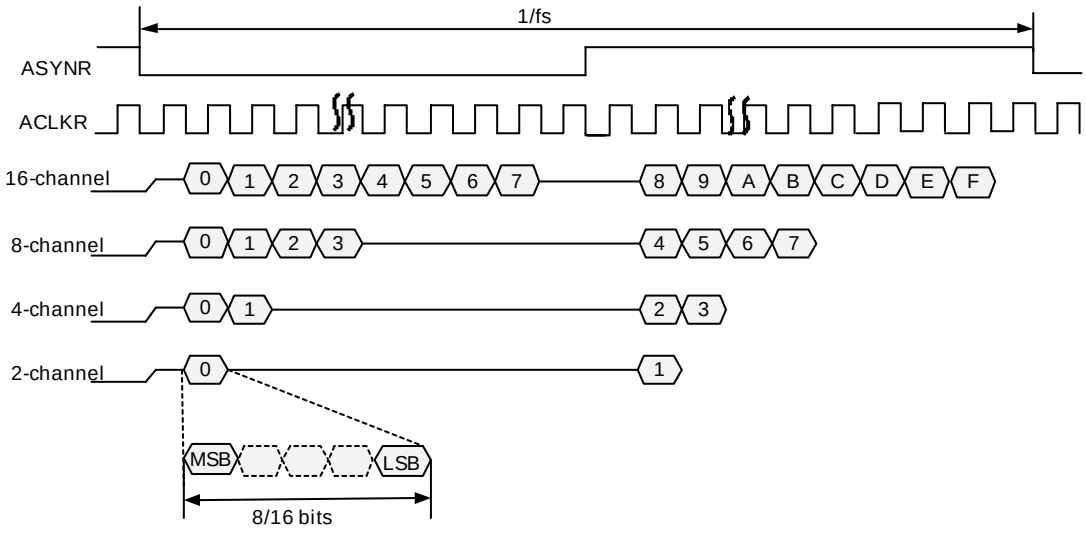
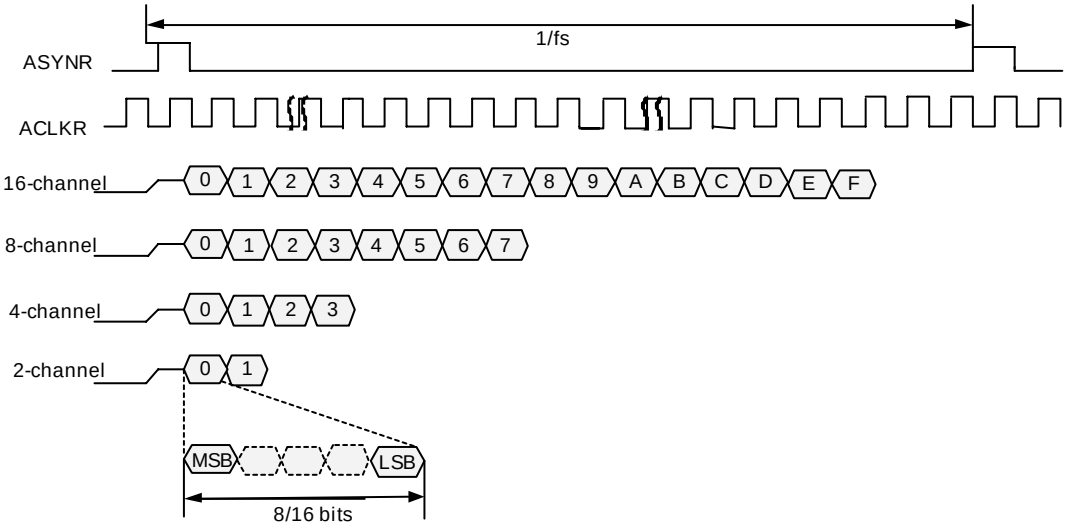


Figure 7-9 shows how data is received in PCM mode. Both the standard and customized PCM modes are supported. Unlike the single-channel mode, the SIO interface can receive data on the rising edge or falling edging optionally. Figure 7-9 shows how the SIO interface receives data on the rising edge in PCM mode.

Figure 7-9 Receiving 2-/4-/8-/16-channel data through the PCM interface



7.5 Operating Mode

Pin Multiplexing

The SIO pins can be multiplexed by configuring IO config registers reg39 to reg41. For details about pin multiplexing, see Table 7-2.

Table 7-2 Pin multiplexing

Signal Name	Multiplex Signal
SIO0XFS	GPIO0_2
SIO0XCK	GPIO0_3
ACKOUT	GPIO0_4

Clock Gating

If no audio recording or playing is required after SIO_CT_SET[rx_enable] and SIO_CT_SET[tx_enable] of SIO0 or SIO1 are set to 0, you can disable the SIO clocks by writing 1 to the related bits of the write-only SC_PERDIS register. Writing 0 has no effect. You can disable the clocks of SIO0 and SIO1 as follows:

- Write 1 to SC_PERDIS[sio0_clkdis] to disable the clock of SIO0.
- Write 1 to SC_PERDIS[sio1_clkdis] to disable the clock of SIO1.

You can enable the clocks of SIO0 and SIO1 by writing 1 to the related bits of the SC_PEREN register as follows:

- Write 1 to SC_PEREN[sio0clken] to enable the clock of SIO0.
- Write 1 to SC_PEREN[sio1clken] to enable the clock of SIO1.

Whether the clocks of the SIO interfaces are enabled can be checked by reading the related bits of the SC_PERCLKEN register. The value 0 indicates that the related clock is disabled; the value 1 indicates that the related clock is enabled.

Clock Configuration

The two SIO interfaces work independently, but their clocks are configured in the same manner.

If SIO0 or SIO1 is required to work in master mode, you need to set SC_PERIPHCTRL12[sio0_master] or SC_PERIPHCTRL12[sio1_master] to 1. After that, you need to select the frequency ratio of the bit clock and sync clock by configuring SC_PERIPHCTRL13 and SC_PERIPHCTRL14.

Soft Reset

SIO0 and SIO1 can be independently reset by setting SC_PERIPHCTRL8[sio0_srst] and SC_PERIPHCTRL8[sio1_srst] to 1. After the reset, each configuration register is restored to its default value. Therefore, these registers must be reinitialized.

Audio Playing and Recording in Interrupt/Query Mode

1. Initialization

The initialization procedure is as follows:

Step 1 Set SIO_CT_SET/SIO_CT_CLR[rx_enable] and SIO_CT_SET/SIO_CT_CLR[tx_enable] to 0 to disable the SIO.

Step 2 Select the I^2S or PCM mode by setting SIO_MODE[sio_mode]. If you choose the PCM mode, select the timing type by setting SIO_MODE[pcm_mode]. If multi-channel audio recording is required, set SIO_MODE[ext_rec_en], SIO_MODE[chn_num], and SIO_MODE[clk_edge].

Step 3 If the SIO works in master mode, configure the clock frequency. If the SIO works in slave mode, no configuration is required.

Step 4 Set the bit width by configuring SIO_DATA_WIDTH_SET and SIO_SIGNED_EXT.

Step 5 Set the thresholds for the RX_FIFO and TX_FIFO by configuring SIO_CT_SET[rx_fifo_threshold] and SIO_CT_SET[tx_fifo_threshold].

Step 6 In I^2S mode, configure SIO_I2S_POS_MERGE_EN and SIO_I2S_START_POS according to how data is read from or written into the FIFO. At the same time, configure SIO_CT_SET[tx_data_merge_en] and SIO_CT_SET[rx_data_merge_en]. For multi-channel audio recording, skip this step.

Step 7 Set the SIO interrupt mask register SIO_INTMASK and SIO_CT_SET[intr_en] as required.

Step 8 Configure the external audio codecs.

----End

2. Audio Playing

To play audios, do as follows:

Step 1 Set SIO_CT_SET[tx_fifo_disable] to 1 and then to 0 to clear the data remained in the TX_FIFO.

Step 2 Write the data to be transmitted to the TX_FIFO, and then write 1 to SIO_CT_SET[tx_enable] to start data transmit.

Step 3 In query mode, check the status of TX_FIFO by reading SIO_TX_STA; in interrupt mode, report an interrupt according to the interrupt status indicated by SIO_INTSTATUS[tx_intr]. When the data depth of the TX_FIFO is detected below the threshold, write data into the TX_FIFO until data is transmitted completely. When data is transmitted completely, go to Step 4. Ensure that data underflow does not occur in the TX_FIFO before data is transmitted completely. Otherwise, the sound may be discontinuous.

Step 4 Set SIO_CT_SET[tx_enable] to 0.

----End

3. Audio Recording

To record audios, do as follows:

Step 1 Set SIO_CT_SET[rx_fifo_disable] to 1 and then to 0 to clear the data remained in the RX_FIFO.

Step 2 Set SIO_CT_SET[rx_enable] to 1 to start data receive.

Step 3 In query mode, check the status of the RX_FIFO by reading SIO_RX_STA; in interrupt mode, check the status of the RX_FIFO through the related interrupt status bit. When the data depth of the RX_FIFO is detected above the threshold, read data from the RX_FIFO until data is received completely. When data is received completely, go to Step 4. Ensure that data overflow does not occur in the RX_FIFO before data is transmitted completely. Otherwise, data loss occurs.

Step 4 Set SIO_CT_SET[rx_enable] to 0 to read all the data remained in the RX_FIFO.

----End

Audio Playing and Recording in DMA Mode

1. Initialization

The audio playing and recording in DMA mode is performed in the same way as that in interrupt/query mode.

2. Audio Playing

To play audios, do as follows:

Step 1 Set the interrupt mask register SIO_INTMASK[tx_intr] to 1 to mask transmit interrupts.

Step 2 Configure the DMA data channel, including the data transmission source, destination address, amount of data to be transmitted, and transmission type. For details, see the configuration description of DMA.

Step 3 Set SIO_CT_SET[tx_fifo_disable] to 1 and then to 0 to clear the data remained in the TX_FIFO.

Step 4 Write the initial data into the TX_FIFO to enable the depth to be above the FIFO threshold. For example, you can write the data 0x0, which indicates mute. If no initial data is written into the FIFO, the SIO reports a FIFO underflow but the DMA does not write data into the FIFO yet when audio playing is enabled. If the initial data is written into the FIFO, the FIFO underflow upon the start of the audio playing can be prevented.

Step 5 Set SIO_CT_SET[tx_enable] to 1 to start audio playing.

Step 6 Check whether the data is transmitted completely according to the DMA interrupt. If yes, set SIO_CT_SET[tx_enable] to 0.

----End

3. Audio Recording

To record audios, do as follows:

Step 1 Configure the DMA data channel, including the data transmission source, destination address, amount of data to be transmitted, and transmission type. For details, see the configuration description of DMA.

Step 2 Set SIO_CT_SET[rx_fifo_disable] to 1 and then to 0 to clear the data remained in the RX_FIFO.

Step 3 Set SIO_CT_SET[rx_enable] to 1 to start data receive.

Step 4 If you want to stop audio recording, set SIO_CT_SET[rx_enable] to 0.

----End

7.6 Register Summary

The register base addresses of the SIO module are as follows:

- SIO0: 0x1004_0000
- SIO1: 0x1005_0000

Table 7-3 lists the SIO registers.

Table 7-3 Summary of the SIO registers

Offset Address	Register	Description	Page
0x03C	SIO_VERSION	SIO version register	7-12
0x040	SIO_MODE	SIO mode register	7-13
0x044	SIO_INTSTATUS	SIO interrupt status register	7-14
0x048	SIO_INTCLR	SIO interrupt clear register	7-16
0x04C	SIO_I2S_LEFT_XD	I^2S left channel data transmit register	7-17
0x050	SIO_I2S_RIGHT_XD	I^2S right channel data transmit register	7-17
0x050	SIO_PCM_XD	PCM data transmit register	7-18
0x054	SIO_I2S_LEFT_RD	I^2S left channel data receive register	7-18
0x058	SIO_I2S_RIGHT_RD	I^2S right channel data receive register	7-19
0x058	SIO_PCM_RD	PCM data receive register	7-20
0x05C	SIO_CT_SET	I^2S/PCM control set register	7-20
0x060	SIO_CT_CLR	I^2S/PCM control clear register	7-22
0x064	RESERVED	Reserved	-
0x068	SIO_RX_STA	SIO receive status register	7-24
0x06C	SIO_TX_STA	SIO transmit status register	7-25
0x070–0x074	RESERVED	Reserved	-
0x078	SIO_DATA_WIDTH_SET	I^2S/PCM data width set register	7-25
0x07C	SIO_I2S_START_POS	Data access start position control register of I^2S left and right channels	7-26
0x080	I2S_POS_FLAG	Current position status register of I^2S left and right channels	7-27

Offset Address	Register	Description	Page
0x084	SIO_SIGNED_EXT	Upper-bit sign extend enable register	7-28
0x088	SIO_I2S_POS_MERGE_EN	Access position merging enable register of I^{2}S left and right channels	7-28
0x08C	SIO_INTMASK	SIO interrupt mask register	7-29
0x090–0x09C	RESERVED	Reserved	-
0x0A0	SIO_I2S_DUAL_RX_CHN	Data receive register after the enabling of I^{2}S left and right channel data access position merging	7-30
0x0C0	SIO_I2S_DUAL_TX_CHN	Data transmit register after the enabling of I^{2}S left and right channel data access position merging	7-31

7.7 Register Description

SIO_VERSION

SIO_VERSION is the SIO version register. It is used to record the SIO version number and perform the SIO self-test.

offset Address	Register Name	Total Reset Value
0x03C	SIO_VERSION	0x0000_0013

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

Bits	Access	Name	Description
[31:9]	-	reserved	Reserved.
[8]	RW	sio_loop	SIO loop or normal mode select. 0: Normal mode. 1: Data transmit and receive loop mode. It is used for the self-test of the SIO. In the mode, the SIO receive serial data line is directly connected to the SIO transmit serial data line at the external interface of the SIO.

[7:0]	RO	version	Version number of the SIO.

SIO_MODE

SIO_MODE is the SIO mode register. It is used to select the basic operating mode of the SIO as follows:

- In master mode, the clock and sync signals to the codec and SIO are sent by the clock reset generation (CRG) module.
- In slave mode, the clock and sync signals are sent to the SIO by the external codec.

The master or salve mode of the I^{2}S/PCM interface can be selected through the system control register SC_PERCTRL12. For details, see the description of SC_PERCTRL12 in section 3.10.5.

offset Address	Register Name	Total Reset Value
0x040	SIO_MODE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									clk_edge	chn_num		ext_rec_en	reserved	pcm_mode	sio_mode
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:7]	-	reserved	Reserved.
[6]	RW	clk_edge	Clock edge on which data is sampled during multi-channel data receive in PCM mode. 0: Falling edge. 1: Rising edge.
[5:4]	RW	chn_num	Number of channels during multi-channel data receive. 00: 2 channels 01: 4 channels 10: 8 channels 11: 16 channels
[3]	RW	ext_rec_en	In standard data receive mode, the I^{2}S interface receives both the left-channel and right-channel data. The PCM interface, however, receives either the right-channel data or left-channel data. For the I^{2}S/PCM interface in multi-channel data receive mode, the number of channels is configurable. Additionally, the bit width of the channels must be 8 bits or 16 bits. 0: Standard data receive mode of the I^{2}S/PCM interface. 1: Extended multi-channel data receive mode of the I^{2}S/PCM interface.

[2]	-	reserved	Reserved. This bit must be set to 0.
[1]	RW	pcm_mode	PCM timing mode. 0: Standard mode. 1: Customized mode.
[0]	RW	sio_mode	PCM/I^2S mode select. 0: I^2S mode. 1: PCM mode.

SIO_INTSTATUS

SIO_INTSTATUS is the SIO interrupt status register.

Take the receive interrupt as an example. When the data depth of the RX_FIFO is above the threshold, the high level is always latched to the interrupt status register and the interrupts are generated continuously. That is, even the CPU clears the interrupt, the interrupt status register is set to 1 in the next clock cycle. Therefore, it is recommended to perform the following steps through the CPU:

Step 1 Write 1 to SIO_CT_CLR[intr_en] to disable the global interrupt enable bit.

Step 2 Read the interrupt status register SIO_INTSTATUS.

Step 3 Perform relevant operations according to the interrupt source.

Step 4 Write 1 to the related bit of SIO_INTCLR to clear the interrupt.

Step 5 Write 1 to SIO_CT_SET[intr_en] to enable the global interrupt enable bit.

----End

The transmit interrupt and receive interrupt are generated in the same manner. Therefore, it is recommended to process the transmit interrupt by following the preceding steps.

This register is a raw interrupt status register. If the related interrupt bit is masked and the interrupt condition is met, the related interrupt status bit is set to 1 but no interrupt is triggered.

Offset Address	Register Name	Total Reset Value
0x044	SIO_INTSTATUS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																										tx_left_fifo_under	tx_right_fifo_under	rx_left_fifo_over	rx_right_fifo_over	tx_intr	rx_intr
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:6]	-	reserved	Reserved.
[5]	RO	tx_left_fifo_under	In I^2S mode, this bit indicates the left channel TX_FIFO underflow interrupt status. In PCM mode, this bit is invalid. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	tx_right_fifo_under	In I^2S mode, this bit indicates the right channel TX_FIFO underflow interrupt status. In PCM mode, this bit indicates the PCM TX_FIFO underflow flag. 0: No interrupt is generated. 1: An interrupt is generated.
[3]	RO	rx_left_fifo_over	In I^2S mode, this bit indicates the left channel RX_FIFO overflow interrupt status. In PCM mode, this bit is invalid. 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	rx_right_fifo_over	In I^2S mode, this bit indicates the right channel RX_FIFO overflow interrupt status. In PCM mode, this bit indicates the PCM RX_FIFO underflow flag. 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	tx_intr	Interrupt status when the TX_FIFO level is below the threshold. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	rx_intr	Interrupt status when the RX_FIFO level is above the threshold. 0: No interrupt is generated. 1: An interrupt is generated.

SIO_INTCLR

SIO_INTCLR is the SIO interrupt clear register. It can be cleared bit by bit.

Offset Address	Register Name	Total Reset Value
0x048	SIO_INTCLR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																										tx_left_fifo_under	tx_right_fifo_under	rx_left_fifo_over	rx_right_fifo_over	tx_intr	rx_intr
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:6]	-	reserved	Reserved.
[5]	WO	tx_left_fifo_under	In I^{2}S mode, this bit controls whether the left channel TX_FIFO underflow interrupt is cleared. In PCM mode, this bit is invalid. 0: Not cleared. 1: Cleared.
[4]	WO	tx_right_fifo_under	In I^{2}S mode, this bit controls whether the right channel TX_FIFO underflow interrupt is cleared. In PCM mode, this bit indicates PCM TX_FIFO underflow interrupt clear. 0: Not cleared. 1: Cleared.
[3]	WO	rx_left_fifo_over	In I^{2}S mode, it controls whether the left channel RX_FIFO overflow interrupt is cleared. In PCM mode, this bit is invalid. 0: Not cleared. 1: Cleared.
[2]	WO	rx_right_fifo_over	In I^{2}S mode, this bit controls whether the right channel RX_FIFO overflow interrupt is cleared. In PCM mode, this bit indicates the PCM RX_FIFO overflow interrupt clear. 0: Not cleared. 1: Cleared.
[1]	WO	tx_intr	This bit controls whether the TX_FIFO level below the threshold interrupt is cleared. 0: Not cleared. 1: Cleared.
[0]	WO	rx_intr	This bit controls whether the RX_FIFO level above the threshold interrupt is cleared. 0: Not cleared.

			1: Cleared.

SIO_I2S_LEFT_XD

SIO_I2S_LEFT_XD is the I^2S left channel data transmit register.

The SIO module places the valid data in the lower-bit area when writing data into the register. For example, when the data width is 8 bits, bit[7:0] are valid and bit[31:8] are invalid; when the data width is 16 bits, bit[15:0] are valid and bit[31:16] are invalid. The bits beyond the valid data width are automatically set to 0s by the SIO module.

offset Address	Register Name	Total Reset Value
0x04C	SIO_I2S_LEFT_XD	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	WO	tx_left_data	Left channel transmit data.

SIO_I2S_RIGHT_XD

SIO_I2S_RIGHT_XD is the I^2S right channel data transmit register. The PCM data transmit register is multiplexed with SIO_I2S_RIGHT_XD.

The SIO module places the valid data in the lower-bit area when writing data into the register. For example, when the data width is 8 bits, bit[7:0] are valid and bit[31:8] are invalid; when the data width is 16 bits, bit[15:0] are valid and bit[31:16] are invalid. The bits beyond the valid data width are automatically set to 0s by the SIO module.

Offset Address	Register Name	Total Reset Value
0x050	SIO_I2S_RIGHT_XD	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	WO	tx_right_data	Right channel transmit data.

SIO_PCM_XD

SIO_PCM_XD is the PCM data transmit register. The PCM data transmit register is multiplexed with SIO_I2S_RIGHT_XD.

The SIO module places the valid data in the lower-bit area when writing the valid data to the register. For example, when the data width is 8 bits, bit[7:0] are valid and bit[31:8] are invalid. When the data width is 16 bits, bit[15:0] are valid and bit[31:16] are invalid. The bits beyond the valid data width are automatically set to 0s by the SIO module.

Offset Address	Register Name	Total Reset Value
0x050	SIO_PCM_XD	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	WO	tx_data	PCM transmit data.

SIO_I2S_LEFT_RD

SIO_I2S_LEFT_RD is the I^2S left channel data receive register.

The SIO module places the received valid data in the low-bit area of the register. For example, when the data width is 8 bits, bit[7:0] are valid and bit[31:8] are invalid; when the data width is 16 bits, bit[15:0] are valid and bit[31:16] are invalid. The bits beyond the valid data width are automatically set to 0s by the SIO module.

Offset Address	Register Name	Total Reset Value
0x054	SIO_I2S_LEFT_RD	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	rx_left_data	I^2S left channel receive data.

Note: When data receive is disabled in I^2S mode , the right-channel data may not be written into the FIFO. In this case, the number of data items in the left-channel FIFO may be greater than that in the right-channel FIFO by 1. Therefore, the data in the left-channel and right-channel FIFOs must be cleared before the CPU enables the next data receive.

SIO_I2S_RIGHT_RD

SIO_I2S_RIGHT_RD is the I^2S right channel data receive register. The PCM data receive register is multiplexed with SIO_I2S_RIGHT_RD.

The SIO module places the received valid data in the low-bit area of the register. For example, when the data width is 8 bits, bit[7:0] are valid and bit[31:8] are invalid; when the data width is 16 bits, bit[15:0] are valid and bit[31:16] are invalid. The bits beyond the valid data width are automatically set to 0s by the SIO module.

Offset Address	Register Name	Total Reset Value
0x058	SIO_I2S_RIGHT_RD	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	rx_right_data	I^2S right channel receive data.

Note: When data receive is disabled in I^2S mode , the right-channel data may not be written into the FIFO. In this case, the number of data items in the left-channel FIFO may be greater than that in the right-channel FIFO by 1. Therefore, the data in the left-channel and right-channel FIFOs must be cleared before the CPU enables the next data receive.

SIO_PCM_RD

SIO_PCM_RD is the PCM data receive register, which is multiplexed with SIO_I2S_RIGHT_RD.

The SIO module places the received valid data in the low-bit area of the register. For example, when the data width is 8 bits, bit[7:0] are valid and bit[31:8] are invalid; when the data width is 16 bits, bit[15:0] are valid and bit[31:16] are invalid. The bits beyond the valid data width are automatically set to 0s by the SIO module.

Offset Address	Register Name	Total Reset Value
0x058	SIO_PCM_RD	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RO	rx_data	PCM receive data.

SIO_CT_SET

To facilitate the bit operation on the SIO control register, the SIO_CT_SET register with the address of 0x05C is provided. When 1 is written to the related bit of the register, the bit is set to 1. Writing 0 has no effect. This is a read/write register.

Offset Address	Register Name	Total Reset Value
0x05C/0x060	SIO_CT_SET	0x0000_8000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																rst_n	intr_en	rx_enable	tx_enable	rx_fifo_disable	tx_fifo_disable	rx_data_merge_en	tx_data_merge_en	rx_fifo_threshold				tx_fifo_threshold			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15]	RW	rst_n	I^2S/PCM channel reset, active low. This bit is used to reset the transmit and receive modules (including the FIFOs) in I^2S/PCM mode, and thus the RX_FIFO and TX_FIFO status registers are changed to 0. This bit does not reset the CPU interface register module.
[14]	RW	intr_en	Global interrupt enable. 0: Disabled. 1: Enabled.
[13]	RW	rx_enable	Receive channel enable. 0: Disabled. 1: Enabled.
[12]	RW	tx_enable	Transmit channel enable. 0: Disabled. 1: Enabled.
[11]	RW	rx_fifo_disable	RX_FIFO disable. 0: Enabled. 1: Disabled.
[10]	RW	tx_fifo_disable	TX_FIFO disable. 0: Enabled. 1: Disabled.

[9]	RW	rx_data_merge_en	Data receive merging enable. This bit is valid only when the data width is 16 bits in I^2S mode. 0: Disabled. 1: Enabled. If this bit is 1, the left-channel and right-channel data is merged into a 32-bit data and then is stored in the FIFO. The left-channel 16 bits occupy the 16 MSBs, and the right-channel 16 bits occupy the 16 LSBs. Therefore, the utilization and buffer capacity of the FIFO is improved. The CPU reads data from the RX_FIFO in the following order. That is, reads the 32-bit data (formed by merging the16-bit left-channel data and 16-bit right-channel data) from the left-channel FIFO, and then reads the 32-bit data from the right-channel FIFO. The CPU reads data in this manner repeatedly.
[8]	RW	tx_data_merge_en	Data transmit merging enable. It is valid only when the data bit width is 16 bits in I^2S mode 0: Disabled. 1: Enabled. If this bit is 1, the left-channel and right-channel data is merged into a 32-bit data and then is stored in the FIFO. The left-channel 16 bits occupy the 16 MSBs, and the right-channel 16 bits occupy the 16 LSBs. Therefore, the utilization and buffer capacity of the FIFO is improved. The CPU writes data into the TX_FIFO in the following order. That is, writes the 32-bit data (formed by merging the16-bit left-channel data and 16-bit right-channel data) to the left-channel FIFO, and then writes the 32-bit data to the right-channel FIFO. The CPU writes data in this manner repeatedly.
[7:4]	RW	rx_fifo_threshold	RX_FIFO threshold. When rx_right_depth ≥ (rx_fifo_threshold + 1), the receive interrupt and DMA request are reported.
[3:0]	RW	tx_fifo_threshold	TX_FIFO threshold. When the tx_right_depth < (tx_fifo_threshold + 1), the transmit interrupt and DMA request are reported.

SIO_CT_CLR

To facilitate the bit operation on the SIO control register, the SIO_CT_CLR register with the address of 0x06 is provided. When 1 is written to the related bit of the register, the bit is set to 1. Writing 0 has no effect. This is a write-only register.

Offset Address	Register Name	Total Reset Value
0x05C/0x060	SIO_CT_SET	0x0000_8000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																rst_n	intr_en	rx_enable	tx_enable	rx_fifo_disable	tx_fifo_disable	rx_data_merge_en	tx_data_merge_en	rx_fifo_threshold				tx_fifo_threshold			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15]	RW	rst_n	I^2S/PCM channel reset, active low. This bit is used to reset the transmit and receive modules (including the FIFOs) in I^2S/PCM mode, and thus the RX_FIFO and TX_FIFO status registers are changed to 0. This bit does not reset the CPU interface register module.
[14]	RW	intr_en	Global interrupt enable. 0: Disabled. 1: Enabled.
[13]	RW	rx_enable	Receive channel enable. 0: Disabled. 1: Enabled.
[12]	RW	tx_enable	Transmit channel enable. 0: Disabled. 1: Enabled.
[11]	RW	rx_fifo_disable	RX_FIFO disable. 0: Enabled. 1: Disabled.
[10]	RW	tx_fifo_disable	TX_FIFO disable. 0: Enabled. 1: Disabled.

[9]	RW	rx_data_merge_en	Data receive merging enable. This bit is valid only when the data width is 16 bits in I^2S mode. 0: Disabled. 1: Enabled. If this bit is 1, the left-channel and right-channel data is merged into a 32-bit data and then is stored in the FIFO. The left-channel 16 bits occupy the 16 MSBs, and the right-channel 16 bits occupy the 16 LSBs. Therefore, the usage and buffer capacity of the FIFO is improved. The CPU reads data from the RX_FIFO in the following order. That is, reads the 32-bit data (formed by merging the16-bit left-channel data and 16-bit right-channel data) from the left-channel FIFO, and then reads the 32-bit data from the right-channel FIFO. The CPU reads data in this manner repeatedly.
[8]	RW	tx_data_merge_en	Data transmit merging enable. It is valid only when the data bit width is 16 bits in I^2S mode 0: Disabled. 1: Enabled. If this bit is 1, the left-channel and right-channel data is merged into a 32-bit data and then is stored in the FIFO. The left-channel 16 bits occupy the 16 MSBs, and the right-channel 16 bits occupy the 16 LSBs. Therefore, the usage and buffer capacity of the FIFO is improved. The CPU writes data into the TX_FIFO in the following order. That is, writes the 32-bit data (formed by merging the16-bit left-channel data and 16-bit right-channel data) to the left-channel FIFO, and then writes the 32-bit data to the right-channel FIFO. The CPU writes data in this manner repeatedly.
[7:4]	RW	rx_fifo_threshold	RX_FIFO threshold. When rx_right_depth ≥ (rx_fifo_threshold + 1), the receive interrupt and DMA request are reported.
[3:0]	RW	tx_fifo_threshold	TX_FIFO threshold. When the tx_right_depth < (tx_fifo_threshold + 1), the transmit interrupt and DMA request are reported.

SIO_RX_STA

SIO_RX_STA is the SIO receive status register.

Offset Address	Register Name	Total Reset Value
0x068	SIO_RX_STA	0x0000_0000

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

Bits	Access	Name	Description
[31:10]	-	reserved	Reserved.
[9:5]	RO	rx_left_depth	Left channel RX_FIFO depth indicator. These bits are valid only in I^{2}S mode.
[4:0]	RO	rx_right_depth	In I^{2}S mode, it indicates the right channel RX_FIFO depth indicator. In PCM mode, it indicates the PCM RX_FIFO depth indicator.

SIO_TX_STA

SIO_TX_STA is the SIO transmit status register.

Offset Address	Register Name	Total Reset Value
0x06C	SIO_TX_STA	0x0000_0000

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

Bits	Access	Name	Description
[31:10]	RO	reserved	Reserved.
[9:5]	RO	tx_left_depth	Left channel TX_FIFO depth indicator. These bits are valid only in I^{2}S mode.
[4:0]	RO	tx_right_depth	In I^{2}S mode, it indicates the right channel TX_FIFO depth indicator. In PCM mode, it indicates the PCM TX_FIFO depth indicator

SIO_DATA_WIDTH_SET

SIO_DATA_WIDTH_SET is the I^{2}S/PCM data width set register.

Offset Address	Register Name	Total Reset Value
0x078	SIO_DATA_WIDTH_SET	0x0000_0009

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

Bits	Access	Name	Description
[31:6]	-	reserved	Reserved.
[5:3]	RW	rx_mode	Configuration bits of the length of the data to be received. 000: 8 bits 001: 16 bits 010: 18 bits 011: 20 bits 100: 24 bits 101: 32 bits 110–111: reserved The data length of 16 bits, 18 bits, 20 bits, 24 bits, and 32 bits is supported in I^2S mode. The data length of 8 bits and 16 bits is supported in PCM mode. For multi-channel data receive, only the data length of 8 bits and 16 bits is supported in I^2S/PCM mode.
[2:0]	RW	tx_mode	Configuration bits of the length of the data to be transmitted. 000: 8 bits 001: 16 bits 010: 18 bits 011: 20 bits 100: 24 bits 101: 32 bits 110–111: reserved The data length of 16 bits, 18 bits, 20 bits, 24 bits, and 32 bits is supported in I^2S mode. The data length of 8 bits and 16 bits is supported in PCM mode.

SIO_I2S_START_POS

SIO_I2S_START_POS is the data access start position control register of I^2S left and right channels.

This register controls whether the initial access starts from the left channel or right channel after the left-channel and right-channel data access address merging is enabled.

Offset Address	Register Name	Total Reset Value
0x07C	SIO_I2S_START_POS	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	-	reserved	Reserved.
[1]	RW	start_pos_write	When writing data into the TX_FIFO: 0: Start the access from the left channel. 1: Start the access from the right channel.
[0]	RW	start_pos_read	When reading data from the RX_FIFO: 0: Start the access from the left channel. 1: Start the access from the right channel.

I2S_POS_FLAG

I2S_POS_FLAG is the current position status register of I^2S left and right channels.

This register controls whether the subsequent access starts from the left channel or the right channel after the left-channel and right-channel data access address merging is enabled.

Offset Address	Register Name	Total Reset Value
0x080	I2S_POS_FLAG	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	-	reserved	Reserved.
[1]	RO	start_pos_write	When writing data into the TX_FIFO: 0: Start the subsequent access from the left channel.

			1: Start the subsequent access from the right channel.
[0]	RO	start_pos_read	When reading data from the RX_FIFO: 0: Start the subsequent access from the left channel. 1: Start the subsequent access from the right channel.

SIO_SIGNED_EXT

SIO_SIGNED_EXT is the upper-bit sign extend enable register. This upper-bit sign extend enable bit is valid only for the received data rather than the transmitted data. The sign extension function is supported by the received data in PCM and I^{2}S modes.

Assume that the received valid data width is 8 bits, 16 bits, 18 bits, 20 bits, or 24 bits and the upper-bit sign extend enable bit is enabled. When the receive data is converted into a 32-bit data, the invalid upper bit of the 32 bits are set to the value corresponding to the MSBs of the receive data and then written into the RX_FIFO.

The following example is based on the data width of 16 bits:

```
if(data_rx[15]==1)
data_rx[31:16]=0xffff;
else
data_rx[31:16]=0x0000;
```

Offset Address	Register Name	Total Reset Value
0x084	SIO_SIGNED_EXT	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	signed_ext_en	Upper-bit sign extend enable. 0: Disabled.. 1: Enabled.

SIO_I2S_POS_MERGE_EN

SIO_I2S_POS_MERGE_EN is the access position merging enable register of I^{2}S left and right channels.

When the FIFOs of the SIO are read/written in DMA mode in I^2S mode, the CPU needs to continuously configure the DMA operation addresses because the left-channel and right-channel data addresses are different. This reduces the CPU efficiency. To improve the CPU efficiency, the access position merging bit of the left-channel and right-channel is used.

When the access position merging bit is enabled, both the right-channel and left-channel data is read through SIO_I2S_DUAL_RX_CHN register and written through the SIO_I2S_DUAL_TX_CHN register.

Offset Address	Register Name	Total Reset Value
0x088	SIO_I2S_POS_MERGE_EN	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RW	merge_en	Data access position merging enable of the I^2S left and right channels. 0: Disabled. 1: Enabled.

SIO_INTMASK

SIO_INTMASK is the SIO interrupt mask register.

Offset Address	Register Name	Total Reset Value
0x08C	SIO_INTMASK	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																										tx_left_fifo_under	tx_right_fifo_under	rx_left_fifo_over	rx_right_fifo_over	tx_intr	rx_intr
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bits	Access	Name	Description
[31:6]	-	reserved	Reserved.

[5]	RW	tx_left_fifo_under	In I²S mode, this bit indicates left channel TX_FIFO underflow interrupt mask. In PCM mode, this bit is invalid. 0: Not masked. 1: Masked.
[4]	RW	tx_right_fifo_under	In I²S mode, this bit indicates right channel TX_FIFO underflow interrupt mask. In PCM mode, this bit indicates PCM TX_FIFO underflow interrupt mask. 0: Not masked. 1: Masked.
[3]	RW	rx_left_fifo_over	In I²S mode, this bit indicates left channel RX_FIFO overflow interrupt mask. In PCM mode, this bit is invalid. 0: Not masked. 1: Masked.
[2]	RW	rx_right_fifo_over	In I²S mode, this bit indicates right channel RX_FIFO overflow interrupt mask. In PCM mode, this bit indicates PCM RX_FIFO underflow interrupt mask. 0: Not masked. 1: Masked.
[1]	RW	tx_intr	Interrupt mask when the TX_FIFO level is below the threshold. 0: Not masked. 1: Masked.
[0]	RW	rx_intr	Interrupt mask when the RX_FIFO level is above the threshold. 0: Not masked. 1: Masked.

SIO_I2S_DUAL_RX_CHN

SIO_I2S_DUAL_RX_CHN is the data receive register after the enabling of I²S left and right channel data access position merging.

Offset Address	Register Name	Total Reset Value
0x0A0	SIO_I2S_DUAL_RX_CHN	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	rx_data	Received data.

SIO_I2S_DUAL_TX_CHN

SIO_I2S_DUAL_TX_CHN is the data transmit register after the enabling of I^2S left and right channel data access position merging.

Offset Address	Register Name	Total Reset Value
0x0C0	SIO_I2S_DUAL_TX_CHN	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	WO	tx_data	Transmitted data.

Contents

Figures

Tables

8 MMC/SD/SDIO Controller

8.1 Overview

The MMC/SD/SDIO controller (hereinafter referred to as MMC) processes read/write operations on the secure digital (SD) card and multi-media card (MMC) card and supports various extended devices such as Bluetooth and WiFi devices through the Secure Digital Input/Output (SDIO) protocol. The MMC controls the devices complying with the following protocols and are downward compatible with the protocols of earlier versions:

- SD mem-version 2.00
- SDIO-version 1.10
- MMC-version 4.2

8.2 Features

The MMC has the following features:

- Supports data transfer in direct memory access (DMA) mode
- Supports two transmit first-in-first-outs (FIFOs) and two receive FIFOs. The size of each FIFO is 16 x 32 bits.
- Supports configurable FIFO threshold. In DMA transfer mode, the burst length is also configurable.
- Supports interrupt alarms in case of FIFO overflow and underflow. In this way, data can be transferred without any errors.
- Supports the cyclic redundancy check (CRC) generation and check for data and commands
- Supports programmable interface clock frequency
- Disables the MMC clock in low-power mode
- Disables the interface clock in low-power mode
- Supports 1- or 4-bit data transfer
- Supports read and write operations on the data blocks with the size of 1 byte to 65535 bytes
- Reads/writes stream data from/to the MMC card
- Supports the SDIO interrupt detection in 1- and 4-bit data transfer modes.

- Supports SDIO suspend and resume operations
- Supports the SDIO read wait operation

8.3 Signal Description

Table 8-1 lists the MMC interface signals.

Table 8-1 MMC interface signals

Signal	Direction	Description	Corresponding Pin
SDIOCK	O	Output interface clock signal. The signal is multiplexed with the video output (VO) pin. For details about the multiplexing configuration information, see section 8.5.1 "Pin Multiplexing."	VOCK
SDIOCMD	I/O	Bidirectional command signal. The signal is multiplexed with the VO pin. For details about the multiplexing configuration information, see section 8.5.1 "Pin Multiplexing."	VODAT0
SDIODAT0	I/O	Bidirectional data signal 0. The signal is multiplexed with the VO pin. For details about the multiplexing configuration information, see section 8.5.1 "Pin Multiplexing."	VODAT1
SDIODAT1	I/O	Bidirectional data signal 1. The signal is multiplexed with the VO pin. For details about the multiplexing configuration information, see section 8.5.1 "Pin Multiplexing."	VODAT2
SDIODAT2	I/O	Bidirectional data signal 2. The signal is multiplexed with the VO pin. For details about the multiplexing configuration information, see section 8.5.1 "Pin Multiplexing."	VODAT3
SDIODAT3	I/O	Bidirectional data signal 3. The signal is multiplexed with the VO pin. For details about the multiplexing configuration information, see section 8.5.1 "Pin Multiplexing."	VODAT4
SDIODETC	I	SDIO/MMC card detection signal. The signal is multiplexed with the VO pin. For details about the multiplexing configuration information, see section 8.5.1 "Pin Multiplexing."	VODAT5

NOTE

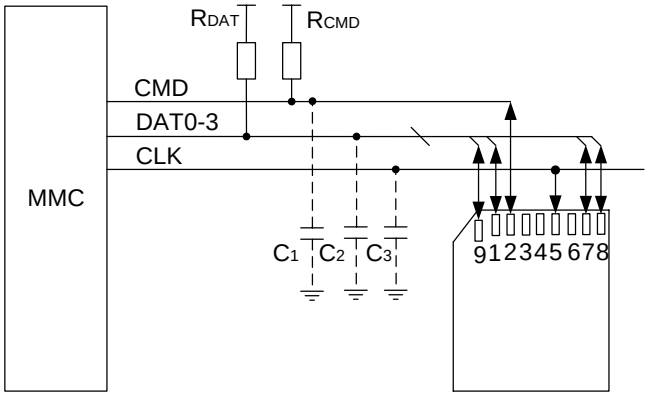
MMC_CCLK, MMC_CMD, and MMC_DAT0–MMC_DAT3 map to the signals CLK, CMD, and DAT0–DAT3 respectively. Those signals are described later.

8.4 Function Description

Typical Application

Figure 8-1 shows the typical application circuit of the MMC.

Figure 8-1 Typical application circuit of the MMC



NOTE

Besides the signal lines shown in Figure 8-1, the card slot also provides a mechanical write-protection signal line and a card detection signal line. By detecting the levels of the two signal lines through the general-purpose input/output (GPIO) pin, the system can implement mechanical write protection and card detection during hot plugging.

By interconnecting with the card device through a clock signal line, a bidirectional command signal line, and four bidirectional data signal lines, the MMC exchanges commands and data with the card device. The command signals and data signals work in pull-up mode. Table 8-2 lists the parameters of the pull-up resistors and the load capacitance of each signal line.

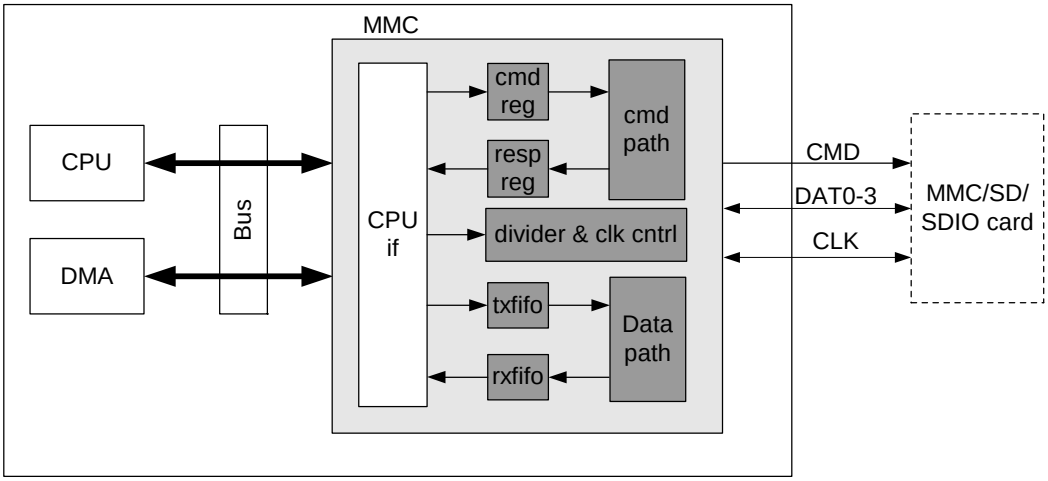
Table 8-2 Load parameters of signal lines

Parameter	Min	Max	Description
R_{DAT}, R_{CMD}	10 kΩ	100 kΩ	Pull-up resistors
Load capacitance Cx	–	40 pF	Load capacitance $C_x = C_{mmchost} + C_{bus} + C_{card}$. The maximum load capacitance (C_{card}) of each card is 10 pF. Hence, $C_{mmchost} + C_{bus} \leq 30$ pF.
Inductance of each signal line	–	16 nH	Fpp ≤ 20 MHz

Functional Principle

Figure 8-2 shows the functional block diagram of the MMC.

Figure 8-2 Functional block diagram of the MMC



The MMC connects to the system through its internal bus. The MMC consists of the command path, data path, and interface clock control unit. Their functions are as follows:

- Command path

  It is used to transmit commands and receive responses.
- Data path

  It is used to perform data read and write operations by working with the command path.
- Interface clock control unit

  It is used to change the frequency of the interface clock as required and enable or disable the interface clock.

Command and Response

All interactions between the MMC and the card device, including card initialization sequence, register read/write, status query, and data transfer, are implemented through commands.

Commands are classified into the following two types according to whether data is transferred:

- Non-data transfer command

  Based on the command signal line CMD, the controller transmits commands and receives responses in serial mode.
- Data transfer command

  Besides the interaction on the command line, data is also transferred through the data lines DAT0 to DAT3.

Commands can also be classified into the following three types according to the response types:

- Command without response

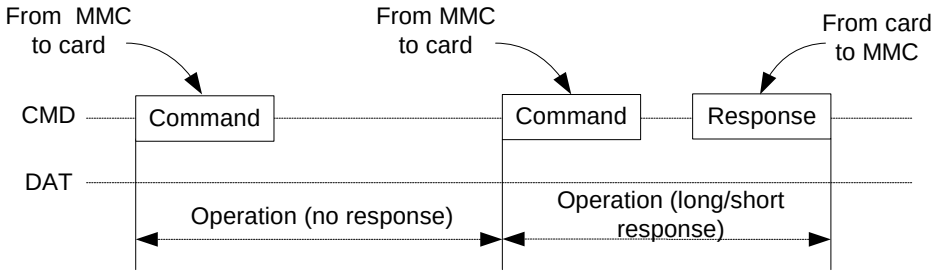
  For example, the card reset command.
- Short response command

  For example, the data transfer command and card status query command.
- Long response command

  This type of commands are only used to read the information about the card identification (CID) and card specific data (CSD) registers of a card.

Figure 8-3 shows the non-data transfer commands between the MMC and the card device. For details about data transfer commands, see Figure 8-6 and Figure 8-7.

Figure 8-3 Non-data transfer commands of the MMC



Both the non-data transfer command and the data transfer command are 48-bit serial data. Each command consists of a start bit, a transfer bit, a command serial number, a command parameter, a CRC bit, and an end bit. After receiving a command, the card returns a 48-bit or a 136-bit response according to the command type. Figure 8-4 shows the command format of the MMC and Figure 8-5 shows the response format of the card device.

Figure 8-4 Command format of the MMC

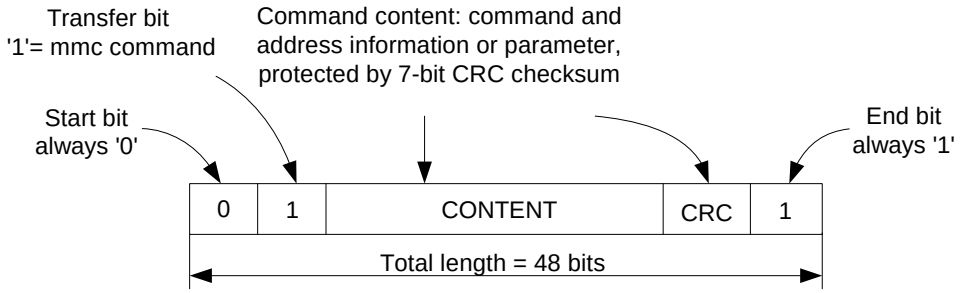
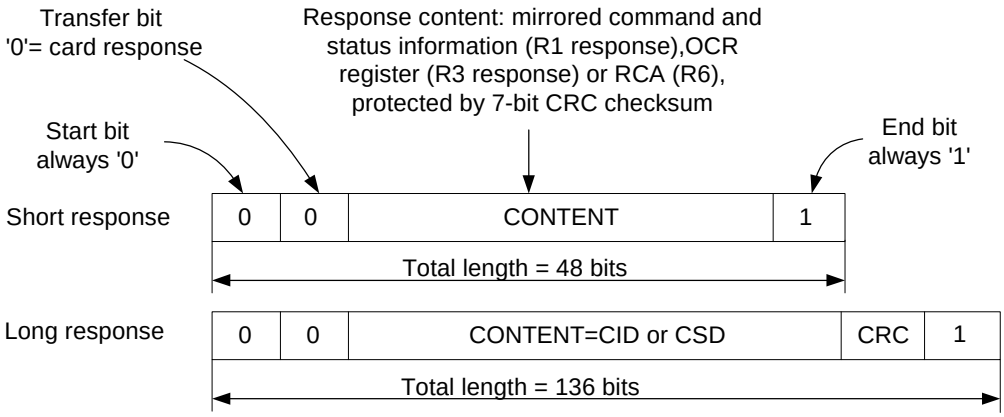


Figure 8-5 Response format of the card device



Data Transfer Command

The MMC, SD card, and SDIO card support the following data read/write modes:

- Stream read/write command mode

  Only the MMC card supports this mode. In this mode, only one data line, namely DAT0, is used for data transfer and no CRC check is performed.

- Single-block read/write command mode

  In this mode, a single block of data is read and written during each data transfer. No stop command is required for stopping each data transfer.

- Multiple-block read/write command mode
  - Predefined block count mode

    Before the multiple block read/write command is executed, the block count command is sent to specify the number of data blocks to be transferred.
  - Open ended mode

    After a read/write command is sent, a stop command is required for stopping data transfer at the end of data transfer.

  The difference between the two modes lies on how the MMC notifies the card of the end of each data transfer. The SD card supports the open-ended mode only, whereas the MMC card supports both modes.

  The multiple block read/write command of 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 sent.

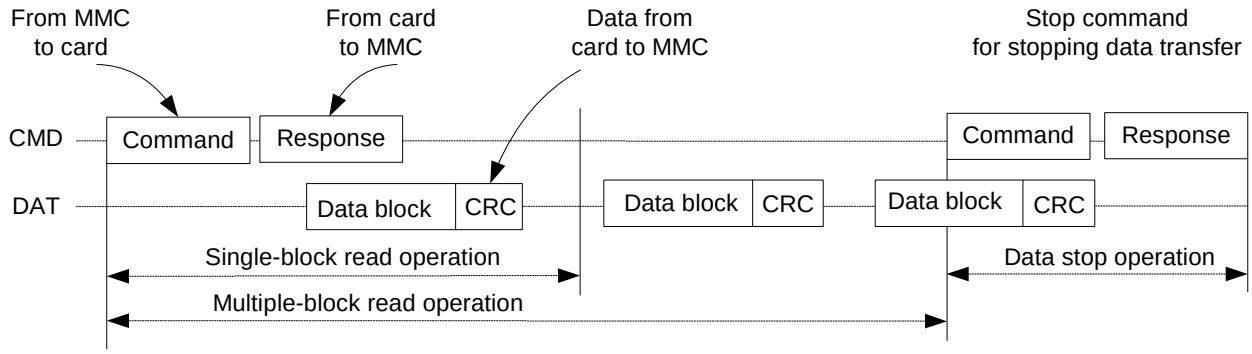
The single-block read/write command mode and the multiple-block read/write command mode are widely used in data transfer. For the data transfer of the SD card and MMC card, a data block is 512 bytes; for the data transfer of the SDIO card, the block size can be customized.

NOTE

In the data transfer through the block read/write command, the total amount of transferred data must be an integer multiple of the block size.

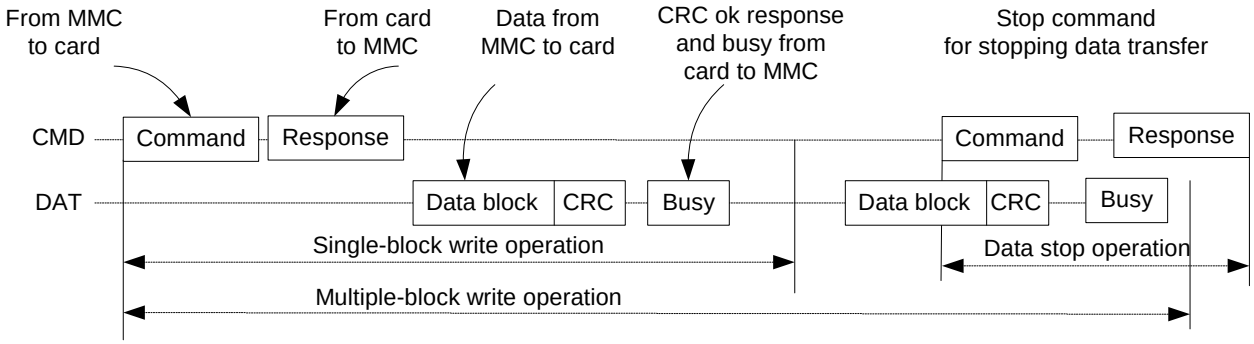
All data transfer commands are short response commands with data transferred through data lines. Figure 8-6 and Figure 8-7 show the relationships between the commands, responses, and timings of data lines.

Figure 8-6 Single-block and multiple-block read operations



The MMC sends a single-block or a multiple-block read command to the card. When receiving the response from the card, the MMC starts to receive data block by block. Each block of data contains a CRC bit that ensures the integrity of data transfer. When sending a single-block read command, the MMC completes a data transfer after a block of data is received. When sending a multiple-block read command in open-ended mode, the MMC completes a data transfer after multiple blocks of data is received and a stop command is sent.

Figure 8-7 Single-block and multiple-block write operations



The MMC sends a single-block or a multiple-block write command to the card. After receiving the response, the MMC starts to transmit data to the card block by block. Each data block contains a CRC bit. Therefore, the card performs CRC on each data block and sends the CRC status to the MMC, thus ensuring that data is transferred properly. For the single-block write operation, the MMC completes a data transfer after a block of data is received. For the multiple-block write operation in open-ended mode, the MMC completes a data transfer after multiple blocks of data is received and a stop command is sent. After a write operation, the card may be busy in programming the flash memory. In this case, the MMC 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.

During the block read/write operations, the 1- or 4-bit data line can be used to transfer data between the controller and the card.

Before a data transfer command is sent, the data transfer widths for the MMC and card must be set to the same value such as 1 bit or 4 bits. You can set the data bit width of the MMC by configuring MMC_CTYPE and set the data bit width of the card by sending the

corresponding command. Figure 8-8 shows the block data format in 1-bit transfer mode and Figure 8-9 shows the block data format in 4-bit transfer mode.

Figure 8-8 Block data format in 1-bit transfer mode

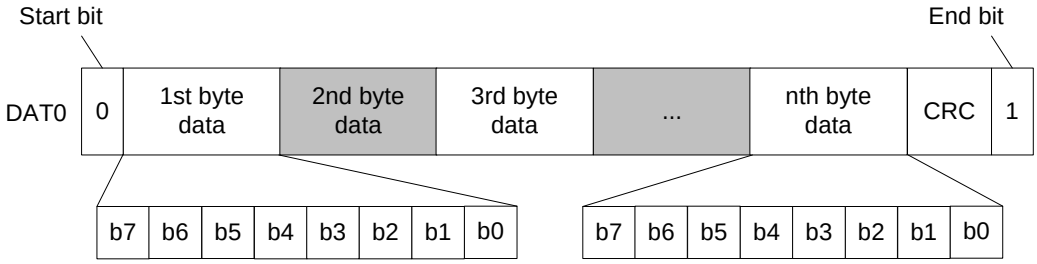
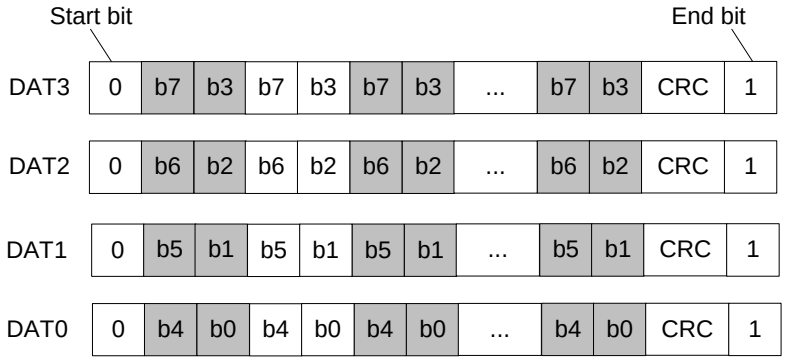


Figure 8-9 Block data format in 4-bit transfer mode



8.5 Operating Mode

8.5.1 Pin Multiplexing

The external pins of the MMC are multiplexed with the GPIO. Therefore, before using the MMC, you need to enable the MMC function of the corresponding pins by configuring IO config registers. For details, see the configuration description of reg28 to reg34.

8.5.2 Clock Gating

When the software completes the current command or data transfer and does not start a new data transfer, the MMC clock can be disabled if the MMC is idle.

To disable the MMC clock, do as follows:

Step 1 Read the MMC_STATUS register.

Step 2 If MMC_STATUS[7:4] and MMC_STATUS[10] are 0, write 0 to MMC_CTRL to mask the controller interrupt and the DMA request enable interrupt, and then go to Step 3. If either MMC_STATUS[7:4] or MMC_STATUS[10] is not 0, wait and then go to Step 1.

Step 3 Set SC_PERDIS[mmcclkdis] to 1 to disable the controller clock.

Step 4 If you want to enable the clock again, write 1 to SC_PEREN[mmcclken].

----End

8.5.3 Soft Reset

When the MMC fails to return to the idle state due to the exception occurred during data transfer, you can soft-reset the MMC by configuring SC_PERCTRL8[mmc_srst]. By default, the reset value 1 indicates that soft reset is cleared. In addition, you can query MMC_STATUS[data_fsm_busy] to check whether the MMC is idle. It is recommended to soft-reset the MMC after a card is inserted during hot-plugging.

8.5.4 Clock Configuration

Configuring the Working Clock

Before the MMC is used, a proper working clock frequency needs to be configured. The working clock of the MMC is generated by dividing the frequency of the PLL clock. The frequency relationship between working clock of the MMC and the PLL output clock is as follows:

$F_{MMMCLK} = F_{PLL}/[(2 \times mmcclk\_sel) + 8]$

Where, mmcclk_sel indicates the frequency divider that is equal to the configured value of SC_PERCTRL9[mmcclk_sel]; F_{MMMCLK} indicates the frequency of MMMCLK that must be less than 50 MHz.

Configuring the Interface Clock

The clock frequencies vary according to the status of cards that comply with different protocols. Therefore, the MMC provides an internal even frequency-division device that generates proper interface clock frequencies by dividing the frequency of the working clock. The frequency relationship between the working clock MMMCLK and the interface clock MMC_CCLK of the MMC is as follows:

$F_{MMC\_CCLK} = F_{MMCCLK}/(2 \times clk\_divider)$

Where, clk_divider indicates the frequency divider that it is equal to the configured value of MMC_CLKDIV[clk_divider].

Before changing the clock frequency of a card, ensure that no data or command is being transferred for any program. To guarantee a glitch-free output clock for the MMC, do as follows when changing the clock frequency of the card:

Step 1 Disable the interface clock.

Set the MMC_CLKENA register to 0x0000_0000, set MMC_CMD[start_cmd], MMC_CMD[update_clk_regs_only], and MMC_CMD[wait_prvdata_complete] to 1, and then wait until MMC_CMD[start_cmd] is cleared automatically.

Step 2 Configure the divider.

Configure the MMC_CLKDIV register according to the required clock frequency, set MMC_CMD[start_cmd] and MMC_CMD[update_clk_regs_only] to 1, and then wait until MMC_CMD[start_cmd] is cleared automatically.

Step 3 Enable the interface clock again.

Set the MMC_CLKENA register to 0x0000_0001, set MMC_CMD[start_cmd] and MMC_CMD[update_clk_regs_only] to 1, and then wait until MMC_CMD[start_cmd] is cleared automatically.

----End

CAUTION

- The values of the clock configuration registers MMC_CLKDIV and MMC_CLKENA are loaded only when MMC_CMD[start_cmd] and MMC_CMD[update_clk_only] are set to 1. After the values are loaded successfully, the MMC clears MMC_CMD[start_cmd] automatically. In this case, if a command is being executed, a hardware locked error (HLE) interrupt is generated. If an HLE interrupt is generated, repeat the preceding operations.
- Note that when a command is being executed or data is being transferred, the clock parameters of the card cannot be changed.

8.5.5 Initialization

Before exchanging commands and data between the MMC and the card, you must initialize the MMC. To initialize the MMC, do as follows:

Step 1 Configure the IO config registers to enable the MMC function of the corresponding pins. For details, see section 8.5.1 "Pin Multiplexing."

Step 2 Configure the frequency of the MMC working clock. For details, see the description of Configuring the Working Clock in section 8.5.4 "Clock Configuration."

Step 3 After the card is powered on and the command and data signal lines are pulled up and become stable, soft-reset the MMC. For details, see section 8.5.3 "Soft Reset."

Step 4 Clear interrupts. Set all bits of the MMC_RINTSTS register to 1 to clear the raw interrupt status bit.

Step 5 Configure the interrupt mask register. Set all bits of MMC_INTMASK to 1 to enable all the interrupt sources. If data is transferred in DMA mode, set MMC_INTMASK[txdr_int_mask] and MMC_INTMASK[rxdr_int_mask] to 0 to mask the transmit FIFO data request (TXDR) interrupt and receive FIFO data request (RXDR) interrupt.

Step 6 Set MMC_CTRL[int_enable] to 1 to enable the interrupt function of the MMC. If data is transferred in DMA mode, set MMC_CTRL[dma_enable] to 1 to enable the DMA request function of the MMC.

Step 7 Configure the timeout parameter register MMC_TMOUT.

Step 8 Configure the FIFO parameter register MMC_FIFOTH, including the burst size, receive threshold rx_wmark, and transmit threshold tx_wmark during DMA transfer.

----End

After the preceding steps, the interface clock can be configured and commands can be sent to the card.

8.5.6 Non-Data Transfer Commands

After sending a command, the MMC sets MMC_RINTSTS[cmd_done] to 1 if it receives any response such as the error response, valid response, and response timeout (RTO). Generally, short responses are stored in the MMC_RESP0 register and long responses are stored in the registers MMC_RESP0 to MMC_RESP3. MMC_RESP3[31] is the most significant bit (MSB), whereas MMC_RESP0[0] is the least significant bit (LSB). After a command is sent, its error status is reflected by the response command and the corresponding error bit of the MMC_RINTSTS register.

To send a non-data transfer command, do as follows:

Step 1 Configure the corresponding command parameters of MMC_CMDARG.

Step 2 Configure the MMC_CMD register. For details, see Table 8-3.

Step 3 Wait for the execution of the command by the MMC. If the command is executed successfully, the MMC clears MMC_CMD[start_cmd] automatically.

Step 4 Query MMC_RINTSTS[hle_int_status] to check whether an HLE interrupt is generated.

Step 5 Wait until the command is executed. The MMC sets MMC_RINTSTS[cmd_done] to 1 regardless of whether the MMC receives a response or the response times out.

Step 6 Check whether there is any response exception and read the response value if necessary. Read the registers MMC_RINTSTS[rto_int_status], MMC_RINTSTS[rcrc_int_status], and MMC_RINTSTS[re_int_status] to check for RTO, response CRC error (RCRC), or response error (RE).

----End

CAUTION

The values of the command registers MMC_BYTCNT, MMC_BLKSIZ, MMC_CMDARG, and MMC_CMD can be loaded only when MMC_CMD[start_cmd] is set to 1 and MMC_CMD[update_clk_only] is set to 0. After the values are loaded successfully, the MMC clears MMC_CMD[start_cmd] automatically. If a command is being executed, an HLE interrupt is generated. In this case, repeat the preceding operations. When a non-data transfer command is executed, the values of MMC_BYTCNT and MMC_BLKSIZ are ignored.

Table 8-3 Reference configuration of MMC_CMD for non-data transfer command

Parameter	Value	Description
start_cmd	1	Command transmit start bit
update_clk_regs_only	0	Non-clock parameter update command
data_expected	0	Non-data transfer command
cmd_index	Cmd index	Command index
send_initialization	0	This bit is set to 1 when the command is a card reset command, for example, CMD0.

Parameter	Value	Description
stop_abort_cmd	0	This bit is set to 1 when the command is a data transfer stop command, for example, CMD12.
rsponse_length	0	This bit is set to 1 when the response is a long response.
rsponse_expect	1	This bit is set to 0 when a command is not responded, for example, CMD0, CMD4, and CMD15.
wait_prvdata_complete	0 or 1	Before sending a command, the MMC must wait until the current data transfer command is executed. It is recommended to set this bit to the fixed value 1 except that the command is used for querying the card status or stopping the current data transfer during data transfer.
check_response_crc	0 or 1	CRC bit that indicates whether the MMC checks responses

8.5.7 Reading Single-Block or Multiple-Block Data

To read single-block or multiple-block data, do as follows:

Step 1 Write 1 to MMC_CTRL[fifo_reset] to reset the FIFO pointer, and then query and wait until this bit is cleared automatically.

Step 2 Write the count of bytes to be transmitted to the MMC_BYTCNT register.

Step 3 Write the block size to the MMC_BLKSIZ register.

Step 4 Write the start address for reading data to the MMC_CMDARG register.

Step 5 Configure the MMC_CMD register according to the parameters listed in Table 8-4. For the SD card and MMC card, run CMD17 or CMD18 to read single-block or multiple-block data; for the SDIO card, run CMD53 to read single-block or multiple-block data. After the MMC_CMD register is configured, the MMC starts to run the command. After the command is sent to the bus, a cmd_done interrupt is generated.

Step 6 Check MMC_RINTSTS[rxdr_int_status] and MMC_RINTSTS[hto_int_status]. If both or either of them is 1, read data from the FIFO by reading the MMC_DATA register, so the MMC can receive the subsequent data. In addition, check data error interrupts by querying MMC_RINTSTS[7], MMC_RINTSTS[9], MMC_RINTSTS[13], and MMC_RINTSTS[15]. If necessary, send a stop command to stop the data transfer through the program.

Step 7 When MMC_RINTSTS[dto_int_status] is 1, it indicates that data transfer is complete. In this case, read the remaining data from the FIFO by reading the MMC_DATA register.

Step 8 If MMC_CMD[send_auto_stop] is set to 1 when the command is executed, the MMC automatically sends a stop command to stop the data transfer. For details, see section 8.5.11 "Auto-Stop Configuration."

----End

Table 8-4 Reference configuration of MMC_CMD during the process of reading single-block or multiple-block data

Parameter	Value	Description
Default		
start_cmd	1	Command transmit start bit
update_clk_regs_only	0	Non-clock parameter update command.
card_number	0	–
send_initialization	0	This bit is set to 1 when the command is a card reset command, for example, CMD0.
stop_abort_cmd	0	This bit is set to 1 when the command is a data transfer stop command, for example, CMD12.
send_auto_stop	0 or 1	For details, see section 8.5.11 "Auto-Stop Configuration."
transfer_mode	0	Transfer by blocks
read_write	0	Read data from the card
rsponse_length	0	All the responses to data commands are short responses.
data_expected	1	Data transfer command
rsponse_expect	1	This bit is set to 0 when a command is not responded, for example, CMD0, CMD4, and CMD15.
cmd_index	Cmd index	Command index
wait_prvdata_complete	0 or 1	Before sending a command, the host must wait until the current data transfer command is executed. It is recommended to set this bit to the fixed value 1 except that the command is used for querying the card status or stopping the current data transfer.
check_response_crc	0 or 1	CRC bit that indicates whether the MMC checks responses

8.5.8 Writing Single-Block or Multiple-Block Data

To write single-block or multiple-block data, do as follows:

Step 1 Write 1 to MMC_CTRL[fifo_reset] to reset the FIFO pointer, and then query and wait until this bit is cleared automatically.

Step 2 Write the count of bytes to be transmitted to the MMC_BYTCNT register.

Step 3 Write the block size to the MMC_BLKSIZ register.

Step 4 Write the start address for storing data to the MMC_CMDARG register.

Step 5 Write data to the FIFO by writing the MMC_DATA register. The FIFO should be written to full at the very beginning.

Step 6 Configure the MMC_CMD register according to the parameters listed in Table 8-5. For the SD and MMC cards, run CMD24 or CMD25 to write single-block or multiple-block data; for the SDIO card, run CMD53 to write single-block or multiple-block data.

Step 7 Check MMC_RINTSTS[txdr_int_status] and MMC_RINTSTS[hto_int_status]. If both or either of them is 1, write to MMC_DATA to fill data to the FIFO. In addition, detect the data error interrupt by querying MMC_RINTSTS[7], MMC_RINTSTS[9], MMC_RINTSTS[13], and MMC_RINTSTS[15]. If necessary, send a stop command to stop the data transfer through the program. When MMC_RINTSTS[dto_int_status] is 1, it indicates that the data transfer is complete.

Step 8 If MMC_CMD[send_auto_stop] is set to 1 when the command is executed, the MMC automatically sends a stop command to stop the data transfer. For details, see section 8.5.11 "Auto-Stop Configuration."

Step 9 Query and wait until the value of MMC_STATUS[data_busy] is changed from 1 to 0.

----End

Table 8-5 Reference configuration of MMC_CMD during the process of writing single-block or multiple-block data

Parameter	Value	Description
Default		
start_cmd	1	Command transmit start bit
update_clk_regs_only	0	Non-clock parameter update command.
card_number	0	–
send_initialization	0	This bit is set to 1 when the command is a card reset command, for example, CMD0.
stop_abort_cmd	0	This bit is set to 1 when the command is a data transfer stop command, for example, CMD12.
send_auto_stop	0 or 1	For details, see section 8.5.11 "Auto-Stop Configuration."
transfer_mode	0	Transfer by blocks
read_write	1	Write data to the card.
rsponse_length	0	All the responses to data commands are short responses.
data_expected	1	Data transfer command
rsponse_expect	1	This bit is set to 0 when a command is not responded, for example, CMD0, CMD4, and CMD15.
cmd_index	Cmd index	–

Parameter	Value	Description
wait_prvdata_complete	0 or 1	Before sending a command, the host must wait until the current data transfer command is executed. It is recommended to set this bit to the fixed value 1 except that the command is used for querying the card status or stopping the current data transfer.
check_response_crc	0 or 1	CRC bit that indicates whether the MMC checks responses

8.5.9 Reading and Writing Stream Data

The processes of reading and writing stream data are the same as those of reading and writing block data, except that the MMC_CMD[transfer_mode] needs to be set to 1. During stream data transfer, the auto-stop function of the MMC is required..

8.5.10 Transferring Data in DMA Mode

Before the data is transferred in DMA mode, MMC_CMD[dma_enable] must be set to 1 to enable the DMA function of the MMC. In addition, MMC_INTMASK[txdr_int_mask] and MMC_INTMASK[txdr_int_mask] must be set to 0 to mask the RXDR interrupt and TXDR interrupt of the MMC.

To transfer data in DMA mode, do as follows:

Step 1 Reset the FIFO pointer.

Step 2 Configure MMC_BYTCNT, MMC_BLKSIZ, MMC_CMDARG, and MMC_CMD to start the data transfer command.

Step 3 Allocate a DMA channel and configure the registers DMAC_CX_SRC_ADDR, DMAC_CX_DEST_ADDR, DMAC_CX_CONTROL, and DMAC_CX_CONFIG of the channel. If a DMA linked list is used, configure DMAC_CX_LLI of the channel and then enable the channel for data transfer.

Step 4 Query and wait until the DMA interrupt is reported. If a DMA interrupt is reported, it indicates that the data transfer is complete.

Step 5 Check for the data transfer error interrupt and data transfer over (DTO) interrupt.

Step 6 Run the stop command and query MMC_STATUS[data_busy].

----End

Take the following precautions during data transfer:

- When the card is read in non-DMA mode, the DTO interrupt is generated only when the data transfer is complete. When all the data in the card is transmitted, some data may still remain in the FIFO. The generation of the Rx_wmark interrupt depends on the count of bytes remained in the FIFO. After a DTO interrupt is detected through software, read all the data remained in the FIFO.
- When the card is read in DMA mode, the DTO interrupt is generated only when the data in all the FIFOs is written to the memory through the DMA module.

- During data transfer, the data amount must be an integer multiple of the data bit width of the FIFO. For example, if the data amount to be written to the card is 15 bytes, 16 bytes need to be transmitted to the FIFO. Or, when the DMA mode is enabled, the DMA module needs to be programmed for the 16-byte data transfer. MMC_BYTCNT, however, can still be set to 15 through the program. In this case, only 15-byte data is transmitted to the card. Similarly, when 15-bit data is read from the card, the host reads 16-byte data from the FIFO.

8.5.11 Auto-Stop Configuration

When multiple-block data is read or written, a stop command is required to stop each data transfer. The stop command can be sent in xx mode or through the auto-stop function of the MMC. The auto-stop function is configured as follows:

During block data transfer, if MMC_CMD[send_auto_stop] is set to 1, the MMC automatically sends a stop command after all the data is transferred, so the card can be resumed to a specific state. The interrupt bit MMC_RINTSTS[auto_cmd_done] indicates whether the stop command is executed. The corresponding response is stored in the MMC_RESP1 register.

The auto-stop function is applied to the following scenarios:

- SD card

  Multiple block read/write, such as CMD18 and CMD25.
- MMC card
  - Stream data read/write.
  - Multiple block read/write in open-ended mode, such as CMD18 and CMD25.
  - The auto-stop function is not required during multiple block read/write in predefined block count mode. Before CMD18 and CMD25 are sent, CMD23 is sent to specify the number of blocks to be transferred.
- SDIO card

  The auto-stop function is not required.

8.5.12 Stopping or Pausing Data Transfer

The stop command is used to break the data transfer between the MMC and the card. The abort command is used to break the I/O data transfer (for SDIO_IOONLY or SDIO_COMBO only). The two commands are used as follows:

- Stop command

  This command can be sent any time during data transfer, because it is used to stop the data transfer. Therefore, you need to set MMC_CMD[5:0] to CMD12, set MMC_CMD[14] to 1, and set MMC_CMD[13] to 0.
- Abort command

  This command is available for SDIO_IOONLY or SDIO_COMBO only. To abort the data transfer, you need to configure the CCCR[ASx] bit of the SDIO card by running the CMD52 command.

8.5.13 Suspend and Resume Operations

An SDIO card can contain up to seven functional devices. The MMC can suspend the data transfer of a device by performing the suspend operation. Thus, the SD bus can be spared for

another device with higher priority. After the device with higher priority transfers data, the MMC can resume the suspended data transfer of the previous device.

The suspend and resume operations are enabled by configuring the corresponding bits of the CCCR register of the SDIO card. Additionally, the CCCR register is written or read by running the CMD52 command.

To perform a suspend operation, do as follows:

Step 1 Query the SBS bit of the CCCR register to check whether the SDIO card supports suspend and resume operations.

Step 2 Query the FSx and bus status (BS) bits of the CCCR register to check whether the functional device to be suspended is transferring data. Note: If the BS bit is 1, it indicates that the device specified by the FSx bit is transferring data.

Step 3 Set the bus release (BR) bit of the CCCR register to 1 to suspend the current data transfer.

Step 4 Check whether the BS bit and BR bit of the CCCR register are cleared. The BS bit is 1 when the data bus is being used, whereas the BR bit is always 1 before the bus is released completely. When both the BR bit and BS bit are 0, the data transfer of the selected device is suspended successfully.

Step 5 If a read operation is suspended, set MMC_CTRL[abort_read_data] to 1 to reset the data transfer function of the MMC after the suspend operation is performed successfully. After reset, the MMC_CTRL[abort_read_data] bit is cleared automatically.

Step 6 Read MMC_TCBCNT to query the number of transferred bytes.

----End

To perform a resume operation, do as follows:

Step 1 Query the transfer status of the card to check whether the bus is idle.

Step 2 If the card is disconnected, run the CMD7 command to select it. The card status can be queried by running the CMD52 or CMD53 command.

Step 3 Check whether the device to be resumed is ready for data transfer. If the RFx bit of the CCCR register is 1, it indicates that the device is ready.

Step 4 Run the CMD52 command to write the device ID to the FSx bit of the CCCR register to resume the data transfer. Enable the controller to start data transfer when sending the CMD52 command. To be specific, write the block size to the MMC_BLKSIZ register and write the amount of remaining data to be transferred to the MMC_BYTCNT register. Table 8-6 shows the configuration of the MMC_CMDARG register. The configuration of the MMC_CMD register is similar to that in block transfer.

Step 5 The data transfer is resumed after the CMD52 command is sent successfully. Read the resume data flag (DF) bit. If this bit is 1, it indicates that data transfer starts when the transfer function is resumed. If this bit is 0, it indicates that no data is ready for transferring.

Step 6 If the DF bit is 0, the MMC generates a data timeout error interrupt a period of time later during the data read operation.

----End

Table 8-6 Reference configuration of MMC_CMDARG during a resume operation

MMC_CMDARG	Value	Description
bit[31]	1	Read/Write flag
bit[30:28]	0, for CCCR access	Functional device ID
bit[27]	1, read after write	Real-time flag
bit[26]	–	–
bit[25:9]	0x0D	Register address
bit[8]	–	–
bit[7:0]	ID of the functional device that is resumed	Write data

8.5.14 Read Wait Operation

The read wait operation is designed for the SDIO card to abort the data transfer of the current functional device by sending related commands. The MMC can determine how long the data transfer is paused as required. To perform the read wait operation, do as follows:

Step 1 Check whether the card supports the read wait operation and read the SRW bit of the CCCR register by running the CMD52 command. If the SRW bit is 1, it indicates that all the functional devices of the card support the read wait operation.

Step 2 If the card supports this operation, set MMC_CTRL[read_wait] to 1.

Step 3 If you want to resume the data transfer, clear MMC_CTRL[read_wait].

----End

8.6 Register Summary

Table 8-7 lists the MMC registers.

Table 8-7 Summary of the MMC registers (base address: 0x1003_0000)

Offset Address	Register	Description	Page
0x000	MMC_CTRL	Control register	8-19
0x004	RESERVED	Reserved	–
0x008	MMC_CLKDIV	Clock divider register	8-21
0x00C	RESERVED	Reserved	–
0x010	MMC_CLKENA	Clock enable register	8-21
0x014	MMC_TMOUT	Timeout parameter register	8-22
0x018	MMC_CTYPE	Interface bit width register	8-23

Offset Address	Register	Description	Page
0x01C	MMC_BLKSIZ	Block size register	8-23
0x020	MMC_BYTCNT	Transfer data byte count register	8-23
0x024	MMC_INTMASK	Interrupt mask register	8-24
0x028	MMC_CMDARG	Command parameter register	8-26
0x02C	MMC_CMD	Command register	8-26
0x030	MMC_RESP0	Command response register 0	8-28
0x034	MMC_RESP1	Command response register 1	8-28
0x038	MMC_RESP2	Command response register 2	8-29
0x03C	MMC_RESP3	Command response register 3	8-29
0x040	MMC_MINTSTS	Interrupt status register	8-30
0x044	MMC_RINTSTS	Raw interrupt status register	8-31
0x048	MMC_STATUS	Status register	8-34
0x04C	MMC_FIFOTH	FIFO parameter register	8-35
0x050	MMC_CARDSTATUS	Card detection status register	8-37
0x054–0x058	RESERVED	Reserved	–
0x05C	MMC_TCBCNT	Interface transfer count register	8-37
0x060	MMC_TBBCNT	FIFO transfer count register	8-38
0x064–0xFF	RESERVED	Reserved	–
0x100	MMC_DATA	Data register	8-38

8.7 Register Description

MMC_CTRL

MMC_CTRL is the MMC control register. It controls the MMC globally, including interrupt global enable control and DMA enable control.

Offset Address	Register Name	Total Reset Value
0x000	MMC_CTRL	0x0000_0000

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

Name	reserved	abort_read_data	send_irq_response	read_wait	dma_enable	int_enable	reserved	dma_reset	fifo_reset	reserved
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0									

Bits	Access	Name	Description
[31:9]	-	reserved	Reserved. These bits are 0 by default. The value 1 cannot be written to these bits.
[8]	RW	abort_read_data	Reset control bit by reading the state machine. 0: default hold value 1: If the suspend operation is performed to suspend the ongoing data read transfer, this bit is enabled through software after the suspend operation is complete. In this way, the MMC is restored to the idle state from the data transfer state (waiting for the next data block). After the MMC is restored to the idle state, this bit is cleared automatically.
[7]	RW	send_irq_response	0: default hold value 1: An auto interrupt request (IRQ) response is sent. This bit is automatically cleared after the response is sent. To wait for the MMC interrupt, the MMC sends the CMD40 response and waits for the interrupt response from the MMC. In addition, if you do not want the MMC to keep in the interrupt wait state, enable this bit to send the CMD40 response. Then, the MMC is restored to the idle state.
[6]	RW	read_wait	0: clear read wait 1: enable read wait This bit indicates that a read wait command is sent to the SDIO card.
[5]	RW	dma_enable	DMA transfer mode enable bit. 0: disabled 1: enabled Even if the DMA mode is enabled, the CPU can still read or write to the FIFO. In practice, this situation should be avoided, because if the DMA and CPU read and write to the FIFO at the same time, the MMC cannot arbitrate their priorities.
[4]	RW	int_enable	Global interrupt enable bit. 0: disabled 1: enabled The interrupt output is valid only when this bit is valid and the interrupt source is enabled.
[3]	-	reserved	Reserved.

[2]	RW	dma_reset	Function reset control bit of the DMA interface. 0: not reset 1: reset Write 1 to enable reset. After reset, this bit is cleared automatically.
[1]	RW	fifo_reset	FIFO reset control bit. 0: Do not reset the FIFO read/write pointer. 1: Reset the FIFO read/write pointer. Write 1 to enable reset. After reset, this bit is cleared automatically.
[0]	-	reserved	Reserved. This bit is 0 by default. The value 1 cannot be written to this bit.

MMC_CLKDIV

MMC_CLKDIV is the divider register of the interface clock. It controls the frequency of the interface clock. The frequency relationship between the working clock MMMCLK and the interface clock MMC_CCLK of the MMC is as follows: $F_{MMC\_CCLK} = F_{MMMCLK}/(2 \times clk\_divider)$. The value of this register is loaded only when MMC_CMD[start_cmd] and MMC_CMD[update_clk_only] are set to 1.

Offset Address	Register Name	Total Reset Value
0x008	MMC_CLKDIV	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	-	reserved	Reserved.
[7:0]	RW	clk_divider	Even frequency divider of the interface clock. For example, 0x0 indicates no frequency division, 0x1 indicates that the frequency divider is 2 and 0xFF indicates that the frequency divider is 510.

MMC_CLKENA

MMC_CLKENA is the MMC interface clock enable register. The value of this register is loaded only when MMC_CMD[start_cmd] and MMC_CMD[update_clk_only] are set to 1.

Offset Address	Register Name	Total Reset Value
0x010	MMC_CLKENA	0x0000_0000

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

Name	reserved	cclk_low_power	reserved	cclk_enable
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0

Bits	Access	Name	Description
[31:17]	-	reserved	Reserved.
[16]	RW	cclk_low_power	Low-power control of the card. 0: non-lower-power mode 1: low-power mode. When the card is idle, the MMC disables the interface clock automatically. This function, however, is only applicable to the MMC card and SD card. For the SDIO card, the interface clock cannot be disabled; otherwise, the SIDO interrupt cannot be detected.
[15:1]	-	reserved	Reserved.
[0]	RW	cclk_enable	Clock enable control of the card. 0: disabled 1: enabled

MMC_TMOUT

MMC_TMOUT is the timeout parameter register. It is used to configure the timeout parameters during the data read operation and command response.

Offset Address	Register Name	Total Reset Value
0x014	MMC_TMOUT	0xFFFF_FF40

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

Bits	Access	Name	Description
[31:8]	RW	data_timeout	Read data timeout parameter. It also functions as the data starvation interrupt timeout parameter. It is in the unit of clock cycle and the recommended value is 0xFF_FFFF.
[7:0]	RW	response_timeout	Response timeout parameter. It is in the unit of clock cycle and the recommended value is 0xFF.

MMC_CTYPE

MMC_CTYPE is the interface bit width register. It is used to configure the working bit width mode of the MMC, that is, 1-bit or 4-bit mode. The bit widths of the controller and the card must be the same.

Offset Address	Register Name	Total Reset Value
0x018	MMC_CTYPE	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved. Note that the value 1 cannot be written to bit 16.
[0]	RW	card_width	Bus width of the card interface. 0: 1 bit 1: 4 bits

MMC_BLKSIZ

MMC_BLKSIZ is the block size register. The block size of the SD card or MMC card is 512 bytes, whereas the block size of the SDIO card is customized.

Offset Address	Register Name	Total Reset Value
0x01C	MMC_BLKSIZ	0x0000_0200

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

Bits	Access	Name	Description
[31:16]	-	reserved	Reserved.
[15:0]	RW	block_size	Block size configuration. For example, if these bits are set to 0x0200, it indicates that the block size is 512 bytes.

MMC_BYTCNT

MMC_BYTCNT is the transfer data byte count register. If the count of transferred data bytes is equal to the block size, the data transfer is single-block transfer. If the count of transferred data bytes is equal to an integer multiple n ($n \geq 2$) of the block size, the data transfer is

multiple-block transfer. Note that the count of transferred data bytes must be an integer multiple of the block size.

Offset Address	Register Name	Total Reset Value
0x020	MMC_BYTCNT	0x0000_0200

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

Bits	Access	Name	Description
[31:0]	RW	byte_count	Count of the transferred data bytes. If the bits are set to 0x0200, it indicates that the transferred data is 512 bytes. In block transfer, the value must be set to an integer multiple of the block size.

MMC_INTMASK

MMC_INTMASK is the interrupt mask register. It is used to mask the interrupt requests of the corresponding bits of the MMC_RINTSTS register.

Offset Address	Register Name	Total Reset Value
0x024	MMC_INTMASK	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															sdio_int_mask	ebe_int_mask	acd_int_mask	sbe_int_mask	hle_int_mask	frun_int_mask	hto_int_mask	drto_int_mask	rto_int_mask	dcrc_int_mask	rcrc_int_mask	rxdr_int_mask	txdr_int_mask	dto_int_mask	cd_int_mask	re_int_mask	reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:17]	-	reserved	Reserved.
[16]	RW	sdio_int_mask	SDIO interrupt mask. 0: masked 1: not masked
[15]	RW	ebe_int_mask	End-bit error (EBE) interrupt mask. 0: masked 1: not masked
[14]	RW	acd_int_mask	Auto command done (ACD) interrupt mask. 0: masked 1: not masked
[13]	RW	sbe_int_mask	Start-bit error (SBE) interrupt mask.

			0: masked 1: not masked
[12]	RW	hle_int_mask	HLE interrupt mask. 0: masked 1: not masked
[11]	RW	frun_int_mask	FIFO underrun/overrun error (FRUN) interrupt mask. 0: masked 1: not masked
[10]	RW	hto_int_mask	Data starvation-by-host timeout (HTO) interrupt mask. 0: masked 1: not masked
[9]	RW	drto_int_mask	Data Read Timeout (DRTO) interrupt mask. 0: masked 1: not masked
[8]	RW	rto_int_mask	RTO interrupt mask. 0: masked 1: not masked
[7]	RW	dcrc_int_mask	Data CRC error (DCRC) interrupt mask 0: masked 1: not masked
[6]	RW	rcrc_int_mask	RCRC interrupt mask. 0: masked 1: not masked
[5]	RW	rxdr_int_mask	RXDR interrupt mask. 0: masked 1: not masked
[4]	RW	txdr_int_mask	TXDR interrupt mask. 0: masked 1: not masked
[3]	RW	dto_int_mask	DTO interrupt mask. 0: masked 1: not masked
[2]	RW	cd_int_mask	Command done (CD) interrupt mask. 0: masked 1: not masked
[1]	RW	re_int_mask	RE interrupt mask. 0: masked 1: not masked

[0]	-	reserved	Reserved.

MMC_CMDARG

MMC_CMDARG is the command parameter register. MMC_CMD[5:0] specify the command parameters corresponding to command indexes. For example, during the CMD17 single-block data read operation, MMC_CMD[5:0] = 17 indicates that the MMC_CMDARG register must be set to the card space address. For details about the command parameter corresponding to each command index, see the SD, MMC, and SDIO protocols.

Offset Address	Register Name	Total Reset Value
0x028	MMC_CMDARG	0x0000_0000

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

Bits	Access	Name	Description
31:0	RW	cmd_arg	Configuration of command parameters.

MMC_CMD

MMC_CMD is the command register. It is used to specify the command features, including the command index, response type, whether to perform data transfer, direction of data transfer, and transfer mode.

Offset Address	Register Name	Total Reset Value
0x02C	MMC_CMD	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	start_cmd	reserved									update_clk_reg_only	card_number					send_initialization	stop_abort_cmd	wait_prvdata_complete	send_auto_stop	transfer_mode	read/write	data_transfer_expected	check_repsonse_crc	response_length	response_expect	cmd_index					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	start_cmd	Start bit of command execution or interface clock parameter load. This bit is used together with the update_clk_reg_only bit. When update_clk_reg_only is 0, this bit is set to 1 to start to run commands. When update_clk_reg_only is 1, this bit is set to 0 to start to load interface clock parameters.

			When commands are executed or clock parameters are loaded, this bit is cleared automatically. When the bit is 1, the registers related to the clock and command are not allowed to modify. Otherwise, the HLE interrupt is generated.
[30:22]	-	reserved	Reserved. The value 1 cannot be written to these bits.
[21]	RW	update_clk_reg_only	0: normal command sequence 1: Do not send any command but reload the values of the registers MMC_CLKDIV and MMC_CLKENA that are controlled by the interface clock. If it is not required to send commands to the card, this bit is used to adjust the frequency of the interface clock and disable or enable the interface clock. In normal command sequence, that is, when this bit is set to 0, the values of the following registers are loaded by the MMC: MMC_CMD, MMC_CMDARG, MMC_TMOUT, MMC_CTYPE, MMC_BLKSIZ, and MMC_BYTCNT., Then, the MMC adds the new register values to new commands. If this bit is set to 1, commands are not sent to the card and no CD interrupt is generated.
[20:16]	RW	card_number	Number of the card that is in use. These bits should be set to 0.
[15]	RW	send_initialization	0: Do not transmit the initialization sequence before transmitting a command. 1: Transmit the initialization sequence before transmitting a command (80 clock cycles). After the card is powered on, it takes 80 clock cycles to initialize. Therefore, this bit must be configured when the first command is sent to the card.
[14]	RW	stop_abort_cmd	Reset control bit by reading and writing the state machine. 0: no impact 1: When the stop command is sent to abort the ongoing data transfer abnormally, the MMC is restored from the data transfer state to the idle state if this bit is set to 1.
[13]	RW	wait_prvdata_complete	0: Send a command immediately even if the previous data transfer is not complete. 1: Send a command only when the previous data transfer is complete. If this bit is set to 0 during the transmission of a command, the card state can be read or the data transfer can be aborted during data transfer.
[12]	RW	send_auto_stop	0: Do not send the stop command after the data transfer is complete. 1: Send the stop command after the data transfer is complete. After the MMC is enabled, it sends the stop command automatically at the end of each data transfer. The stop command is not required in predefined block count mode and cmd53 mode. In open-ended transfer mode, however, the stop command is

			necessary. In non-data transfer mode, this bit is ignored.
[11]	RW	transfer_mode	0: This bit is used for the block read/write command. 1: This bit is used for stream data read/write command. In non-data transfer mode, this bit is ignored.
[10]	RW	read/write	0: Read data from the card. 1: Write data to the card. In non-data transfer mode, this bit is ignored.
[9]	RW	data_transfer_expected	0: non-data command 1: data command
[8]	RW	check_repsonse_crc	0: Do not check the command response CRC. 1: Check the command response CRC No valid CRC is returned after some commands are responded. To forbid the MMC to perform CRC, the software needs to disable this function for the preceding commands.
[7]	RW	response_length	0: short response command 1: long response command
[6]	RW	response_expect	0: command without response 1: command with response
[5:0]	RW	cmd_index	Command index

MMC_RESP0

MMC_RESP0 is the command response register 0.

Offset Address	Register Name	Total Reset Value
0x030	MMC_RESP0	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	response0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	response 0	bit[39:8] of a 48-bit short response or bit[31:0] of a 136-bit long response.

MMC_RESP1

MMC_RESP1 is command response register 1.

Offset Address	Register Name	Total Reset Value
0x034	MMC_RESP1	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	response1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	response1	Bit[63:32] of a long response or the response to the auto-stop command. When the MMC transmits the auto-stop command, bit[39:8] of the response are stored in this register. The response to the previous command is still stored in the MMC_RESP0 register. The auto-stop command is used only for data transfer and the corresponding responses are always short responses.

MMC_RESP2

MMC_RESP2 is command response register 2.

Offset Address	Register Name	Total Reset Value
0x038	MMC_RESP2	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	response2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	response2	Bit[95:64] of a long response.

MMC_RESP3

MMC_RESP3 is command response register 3.

Offset Address	Register Name	Total Reset Value
0x03C	MMC_RESP3	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	response3																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description

[31:0]	RO	response3	Bit[127:96] of a long command response.

MMC_MINTSTS

MMC_MINTSTS is the masked interrupt status register.MMC_MINTSTS = MMC_RINTSTS & MMC_INTMASK

An interrupt is reported only when MMC_RINTSTS, its corresponding bits, and MMC_CTRL[int_enable] are 1.

Offset Address	Register Name	Total Reset Value
0x040	MMC_MINTSTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															sdio_int	ebe_int	acd_int	sbe_int	hle_int	frun_int	hto_int	drto_int	rto_int	dcrc_int	rcrc_int	rxdr_int	txdr_int	dto_int	cd_int	re_int	reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:17]	-	reserved	Reserved.
[16]	RO	sdio_int	Masked SDIO interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[15]	RO	ebe_int	Masked EBE interrupt or write no CRC interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[14]	RO	acd_int	Masked ACD interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[13]	RO	sbe_int	Masked SBE interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[12]	RO	hle_int	Masked HLE interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[11]	RO	frun_int	Masked FRUN interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[10]	RO	hto_int	Masked HTO interrupt.

			0: No interrupt is generated. 1: An interrupt is generated.
[9]	RO	drto_int	Masked DRTO interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	rto_int	Masked RTO interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	dcrc_int	Masked DCRC interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[6]	RO	rcrc_int	Masked RCRC interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	rxdr_int	Masked RXDR interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	txdr_int	Masked TXDR interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[3]	RO	dto_int	Masked DTO interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	cd_int	Masked CD interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	re_int	Masked RE interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	-	reserved	Reserved.

MMC_RINTSTS

MMC_RINTSTS is the raw interrupt status register. Writing 1 clears the corresponding bits and writing 0 has no effect. That is, writing 1 clears interrupts.

Offset Address	Register Name	Total Reset Value
0x044	MMC_RINTSTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															sdio_int_status	ebe_int_status	acd_int_status	sbe_int_status	hle_int_status	frun_int_status	hto_int_status	drto_int_status	rto_int_status	dcrc_int_status	rcrc_int_status	rxdr_int_status	txdr_int_status	dto_int_status	cd_int_status	re_int_status	reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:17]	-	reserved	Reserved.
[16]	RW	sdio_int_status	Raw SDIO interrupt. 0: No interrupt is generated. 1: An interrupt is generated.
[15]	RW	ebe_int_status	Raw EBE interrupt. 0: No interrupt is generated. 1: An interrupt is generated if an EBE error occurs during the read operation or no data CRC state or a negative CRC state is returned during the write operation.
[14]	RW	acd_int_status	Raw ACD interrupt. 0: No interrupt is generated. 1: An interrupt is generated when the MMC sends the auto-stop command automatically.
[13]	RW	sbe_int_status	Raw SBE interrupt. 0: No interrupt is generated. 1: SBE interrupt when data is read from the card. In 4-bit mode, if data does not contain a start bit, this interrupt is generated.
[12]	RW	hle_int_status	Raw HLE interrupt. 0: No interrupt is generated. 1: When hardware locks the values of certain registers, the MMC still attempts to write to these registers.
[11]	RW	frun_int_status	Raw FRUN interrupt. 0: No interrupt is generated. 1: An interrupt is generated when the system writes data to a full FIFO or the system reads data from an empty FIFO.
[10]	RW	hto_int_status	Raw HTO interrupt. 0: No interrupt is generated. 1: To avoid data loss, the output clock MMC_CCLK of the MMC is disabled if the FIFO is empty when data is transmitted to the card or the FIFO is full when data is received from the card. After the clock is disabled, the data-starvation counter is enabled. If

			counter overflows and the system does not write data to an empty FIFO or read data from a full FIFO, an interrupt is generated. In this case, the output clock is enabled again only when the system reads and writes to the FIFO.
[9]	RW	drto_int_status	Raw DRTO interrupt. 0: No interrupt is generated. 1: data receive timeout interrupt.
[8]	RW	rto_int_status	Raw RTO interrupt. 0: No interrupt is generated. 1: command response timeout interrupt (no response)
[7]	RW	dcrc_int_status	Raw DCRC interrupt. 0: No interrupt is generated. 1: data receive CRC error interrupt
[6]	RW	rcrc_int_status	Raw RCRC interrupt. 0: No interrupt is generated. 1: command response CRC error interrupt
[5]	RW	rxdr_int_status	Raw RXDR interrupt. 0: No interrupt is generated. 1: An interrupt is generated if the data amount in the FIFO is above the read threshold rx_wmark of the FIFO when data is read from the card. In DMA data transfer mode, this interrupt must be masked.
[4]	RW	txdr_int_status	Raw TXDR interrupt. 0: No interrupt is generated. 1: An interrupt is generated if the data amount in the FIFO is equal to or below the write threshold tx_wmark of the FIFO when data is written to the card. In DMA data transfer mode, this interrupt must be masked.
[3]	RW	dto_int_status	Raw DTO interrupt. 0: No interrupt is generated. 1: data transfer complete interrupt. This interrupt is generated even if an SBE error, a CRC error, or a read data timeout error occurs.
[2]	RW	cd_int_status	Raw CD interrupt. 0: No interrupt is generated. 1: An interrupt is generated after a command is executed and a response is received. This interrupt is generated even if a RE error, an RCRC error, or an RTO error occurs.
[1]	RW	re_int_status	Raw RE interrupt. 0: No interrupt is generated. 1: An interrupt is generated when an error occurs in the received command response.
[0]	-	reserved	Reserved.

MMC_STATUS

MMC_STATUS is the MMC status register. It reflects the working status of the MMC.

Offset Address	Register Name	Total Reset Value
0x048	MMC_STATUS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	dma_req	dma_ack	fifo_count													resp_index						data_fsm_busy	data_busy	data_3_status	cmd_fsm_state				fifo_full	fifo_empty	fifo_tx_watermark	fifo_rx_watermark
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RO	dma_req	DMA request status bit. 0: The MMC does not request for DMA transfer. 1: The controller is requesting for DMA transfer.
[30]	RO	dma_ack	DMA acknowledge status bit. 0: The DMAC does not clear the MMC request. 1: The DMAC clears the MMC request.
[29:17]	RO	fifo_count	Count of existing data in the FIFO, in words.
[16:11]	RO	resp_index	Index of previous response, including the response to the auto-stop command.
[10]	RO	data_fsm_busy	Status bit of the data transmit/receive state machine. 0: The data receive/transmit state machine is idle. 1: The data receive/transmit state machine is busy.
[9]	RO	data_busy	0: The card is idle. 1: The card is busy after data is written or erased. This status bit reflects the inversion of the signal of card data line data[0]. After data is written or erased, query this bit through software. The next operation is allowed only after this bit is changed from 1 to 0.
[8]	RO	data_3_status	The bit reflects the status of the signal of card data line data[3].
[7:4]	RO	cmd_fsm_state	Status bit of the command state machine of the MMC. 0000: idle 0001: send init sequence 0010: Tx cmd start bit 0011: Tx cmd tx bit 0100: Tx cmd index + arg 0101: Tx cmd crc7

			0110: Tx cmd end bit 0111: Rx resp start bit 1000: Rx resp IRQ response 1001: Rx resp tx bit 1010: Rx resp cmd idx 1011: Rx resp data 1100: Rx resp crc7 1101: Rx resp end bit 1110: Cmd path wait NCC 1111: wait, CMD-to-response turnaround
[3]	RO	fifo_full	FIFO full flag bit. 0: not full 1: full
[2]	RO	fifo_empty	FIFO empty flag bit. 0: not empty 1: empty
[1]	RO	fifo_tx_watermark	The count of data in the FIFO is below the threshold tx_wmark.
[0]	RO	fifo_rx_watermark	The count of data in the FIFI is equal to or above threshold rx_wmark.

MMC_FIFOTH

MMC_FIFOTH is the MMC FIFO parameter register. Its recommended value is 0x2007_0008. In DMA data transfer mode, the burst sizes of the DMA and the MMC must be the same. During DMA data transfer, the value of this register cannot be changed.

Offset Address	Register Name	Total Reset Value
0x04C	MMC_FIFOTH	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved		burst_size			rx_wmark											resevered				tx_wmark											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	-	reserved	Reserved.
[30:28]	RW	burst_size	Burst size for a DMA data transfer, in words. The value must be the same as that in DMAC mode. 000: 1 001: 4

			010: 8 Others: reserved The supported combinations of the values of burst_size and tx_wmark are as follows: burst_size = 1, tx_wmark = 1 to 15 burst_size = 4, tx_wmark = 4 burst_size = 4, tx_wmark = 4 burst_size = 4, tx_wmark = 12 burst_size = 8, tx_wmark = 8 burst_size = 8, tx_wmark = 4 The supported combinations of the values of burst_size and rx_wmark are as follows: burst_size = 1, rx_wmark = 0 to 14 burst_size = 4, rx_wmark = 3 burst_size = 4, rx_wmark = 7 burst_size = 4, rx_wmark = 11 burst_size = 8, rx_wmark = 7 burst_size = 8, rx_wmark = 11
[27:16]	RW	rx_wmark	FIFO threshold during the data read operation. When the count of the existing data in the FIFO is above the threshold, a DMA request is raised. If the interrupt is enabled, an interrupt request is generated. In non-DMA mode, the RXDR interrupt is enabled and an interrupt request is raised. If the data count in the FIFI is not above the threshold at the end of the data transfer, no interrupt is generated. If a DTO interrupt is generated, the remaining data is read through software. In DMA mode, if the count of the remaining data is below the threshold at the end of the data transfer, the DMA still reads data in burst mode before the DTO interrupt is generated. The following equation must be met: Tx_wmark ≤ FIFO_DEPTH - 2 The recommended value is (FIFO_DEPTH/2) - 1. That is, a request is raised when the value is greater than [(FIFO_DEPTH/2) – 1].
[15:12]	-	resevered	Reserved.
[11:0]	RW	tx_wmark	FIFO threshold during the data transmit operation. When the count of the existing data in the FIFO is equal to or below the threshold, a DMA request is raised. If the interrupt is enabled, an interrupt request is generated. In non-DMA mode, the TXDR interrupt is enabled and an interrupt request is raised. If the interrupt is generated at the end of the data transfer, the remaining bytes are transmitted through software. In DMA mode, if the remaining data count is less than the burst

			size at the end of the data transfer, the DMA still transmits the remaining data in burst mode. The following equation must be met: Tx_wmark ≥ 1 The recommended value is FIFO_DEPTH/2. That is, a request is raised when the value is equal to or less than FIFO_DEPTH/2.

MMC_CARDSTATUS

MMC_CARDSTATUS is the card detection status register.

Offset Address	Register Name	Total Reset Value
0x050	MMC_CARDSTATUS	0x0000_0001

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RO	card_detctor	Card detection status of the MMC interface. 0: A card is already inserted. 1: No card is inserted.

MMC_TCBCNT

MMC_TCBCNT is the interface transfer count register. It counts the bytes transferred through the interface during a data transfer. During data transfer, the register returns 0. After the data transfer, the register shows the count of bytes transferred between the MMC and the card. When a new data transfer command starts, the register is cleared.

Offset Address	Register Name	Total Reset Value
0x05C	MMC_TCBCNT	0x0000_0000

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

Bits	Access	Name	Description
[31: 0]	RO	trans_card_byte_count	Count of bytes transferred between the MMC and card.

MMC_TBBCNT

MMC_TBBCNT is the FIFO transfer count register. This register counts the bytes transferred between the CPU/DMA and the MMC FIFO in real time when the data transfer command is executed. The byte count is changed dynamically during data transfer. When a new data transfer command starts, the register is cleared.

Offset Address	Register Name	Total Reset Value
0x060	MMC_TBBCNT	0x0000_0000

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

Bits	Access	Name	Description
[31: 0]	RO	trans_fifo_byte_count	Count of bytes transferred between the CPU/DMA and the MMC FIFO.

MMC_DATA

MMC_DATA is the data register for storing the FIFO entrance address. Before the FIFO is read or written, MMC_STATUS[fifo_count] must be read to query the remaining space of the FIFO. Thus, the data bytes to be read or written are determined based on the remaining space. In this way, FIFO overflow is avoided.

Offset Address	Register Name	Total Reset Value
0x100	MMC_DATA	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	fifo_entrance	FIFO entrance address.

Contents

Figures

Tables

9 SATA

9.1 Oveview

The serial advanced technology attachment (SATA) interface of the Hi3515 is an advanced high-performance bus (AHB) interface that complies with advanced microcontroller bus architecture (AMBA) Spec 2.0. This interface is used to implement rapid integration on the system-on-chip (SOC) system. With the drivers developed in Linux, the SATA interface helps software engineers customize and develop the drivers of the SOC subsystem rapidly. The Hi3515 provides two SATA instance interfaces. At the controller layer, the SATA interface supports native command queuing (NCQ), hot-plugging, port multiplier, eSATA, and power supply management.

9.2 Features

The SATA module of the Hi3515 provides the following features:

- Provides the slave interface of the ARM AHB system bus that complies with AMBA Spec 2.0. Only the 32-bit access mode is supported.
- Provides the master interface of the ARM AHB system bus that complies with AMBA Spec 2.0. Only the 32-bit access mode is supported.
- Provides a standard signal interface for connecting to the PHY.
- Supports the protocols SATA2.5 and Advanced Host Controller Interface 1.2 (AHCI1.2).
- Supports programmable input/output (PIO), legacy direct memory access (DMA), NCQ, and advanced technology attachment packet interface (ATAPI) operations.
- Supports power supply management.
- Supports the port multiplier.
- Provides up to two SATA instance ports.
- Supports automatic rate negotiation: 1.5 Gbps or 3.0 Gbps.
- Supports the interrupt report mechanism.
- Meets the requirement for the AMBA bus clock (hclk): 75 MHz ≤ hclk ≤ 200 MHz.

9.3 Signal Description

Table 9-1 lists the signals of the SATA interface.

Table 9-1 Signals of the SATA interface

Signal	Direction	Description	Pin
SREFCKM	I	SATA negative differential clock input	SREFCKM
SREFCKP	I	SATA positive differential clock input	SREFCKP
SRESREF	I/O	SATA extended resistor pin, connected to external extended resistors	SRESREF
SRXM0	I	Negative differential data input of SATA port 0	SRXM0
SRXM1	I	Negative differential data input of SATA port 1	SRXM1
SRXP0	I	Positive differential data input of SATA port 0	SRXP0
SRXP1	I	Positive differential data input of SATA port 1	SRXP1
STXM0	O	Negative differential data output of SATA port 0	STXM0
STXM1	O	Negative differential data output of SATA port 1	STXM1
STXP0	O	Positive differential data output of SATA port 0	STXP0
STXP1	O	Positive differential data output of SATA port 1	STXP1

9.4 Function Description

Typical Applications

Figure 9-1, Figure 9-2, and Figure 9-3 show the typical application modes of the SATA interface.

Figure 9-1 Typical application mode 1

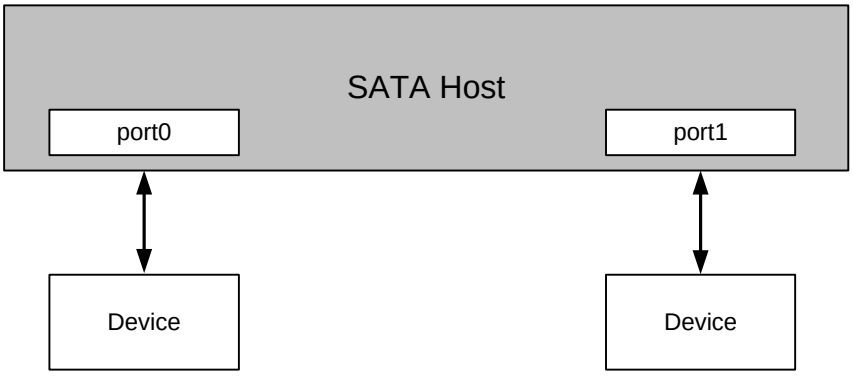


Figure 9-2 Typical application mode 2

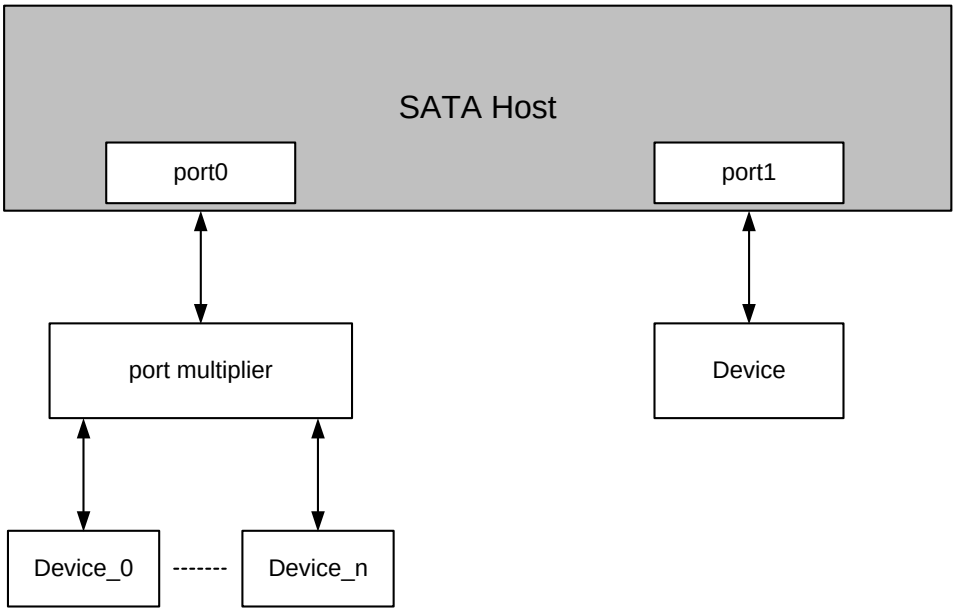
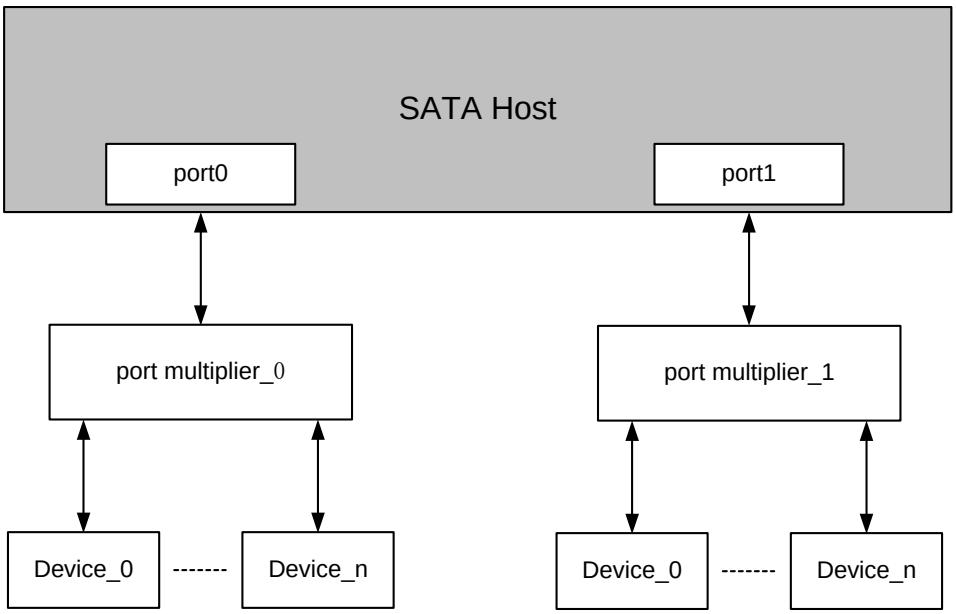


Figure 9-3 Typical application mode 3

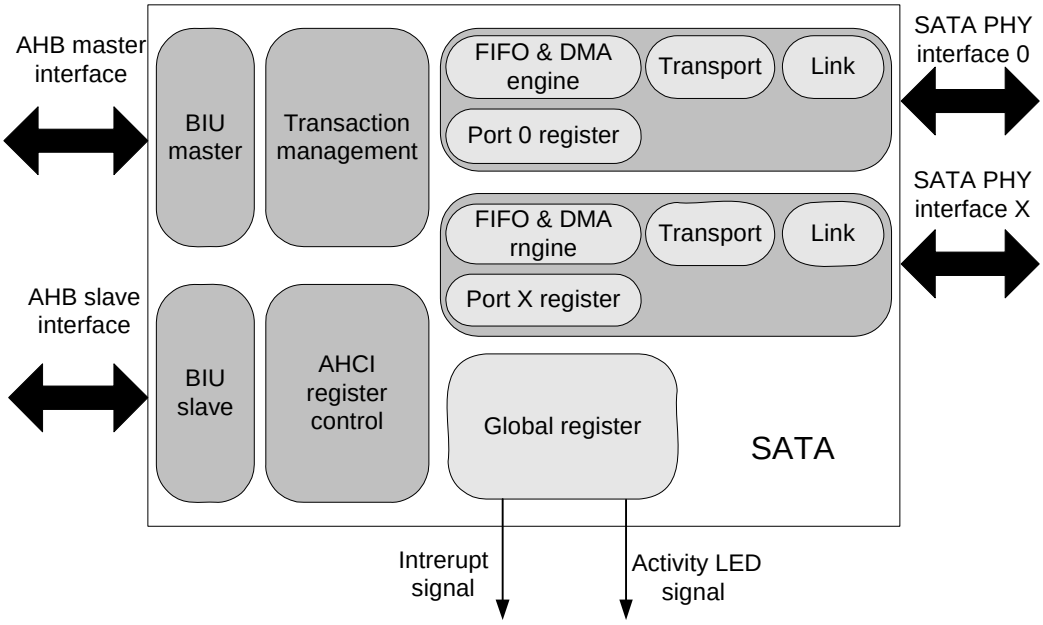


The SATA Host can be directly connected to up to two devices such as the hard disk and driver. In addition, any of the port of the SATA Host can be connected to the port multiplier. The number of connected external devices depends on the expansion capability of the port multiplier.

Function Principle

Figure 9-4 shows the architecture of the Hi3515 SATA module.

Figure 9-4 Architecture of the Hi3515 SATA module



The AHB master and AHB slave interfaces are mounted on the system bus AHB that complies with ARM AMBA2.0. The ARM CPU and system memory controller are also located on the system bus. The software configures the SATA module through the AHB slave interface. Through the AHB master interface, the dynamic random access memory (DRAM) of the system memory controller is accessed to read commands and data and write data.

The SATA Host supports the PIO, legacy DMA, NCQ, and ATAPI operations.

The interrupt signal is connected to the system interrupt controller.

In the Hi3515, the SATA PHY interface is connected to the SATA PHY. The SATA PHY can be connected to the external hard disk or drive with the SATA interface. The SATA PHY can also be connected to the SATA port multiplier for connecting multiple ports.

The activity light emitting diode (LED) signal can be directly transferred from the Hi3515. You can choose this function as required.

9.5 Operating Mode

9.5.1 Clock Gating

You can set SC_PEREN[sataclkgate] to 1 to enable the clock of the SATA module and set SC_PERDIS[sataclkdis] to 1 to disable the clock of the SATA module.

9.5.2 Clock Configuration

You can configure the internal control register SATA_PHY0_CTLH bit[13] of the SATA Host to choose the reference clock source of the SATA PHY. That is, you can control whether the clock of the SATA PHY is generated in the Hi3515 or supplied by the external clock source. The details are as follows:

- When SATA_PHY0_CTLH bit[13] is set to 1, the clocks SREFCKM and SREFCKP are generated in the Hi3515.
- When SATA_PHY0_CTLH bit[13] is set to 0, the clocks SREFCKM and SREFCKP are supplied by the external clock source.

The reference clock supported by the SATA PHY ranges from 25 MHz to 156.25 MHz. The input clock supported by the internal multiplying phase-locked loop (MPLL) of the PHY ranges from 50 MHz to 78.125 MHz. Therefore, when the reference clock is changed, you must configure SATA_PHY0_CTLL bit[31:30] of the SATA Host to meet the clock requirement of the internal MPLL of the SATA PHY. For details, see the description of SATA_PHY0_CTLL bit[31:30].

9.5.3 Soft Reset

The following shows multiple soft reset control modes of the SATA Host:

- Set SC_PERCTRL10 bit[24] to 0 to soft-reset only the sata_alive clock domain.
- Set SC_PERCTRL10 bit[23] to 0 to soft-reset only the sata_rx clock domain of port1.
- Set SC_PERCTRL10 bit[22] to 0 to soft-reset only the sata_rx clock domain of port0.
- Set SC_PERCTRL10 bit[21] to 0 to soft-reset only the sata_tx clock domain of port1.
- Set SC_PERCTRL10 bit[20] to 0 to soft-reset only the sata_tx clock domain of port0.
- Set SC_PERCTRL10 bit[19] to 0 to soft-reset only the SATA PHY.

- Set SC_PERCTRL10 bit[18] to 0 to soft-reset only the HCLK clock domain.
- Set SC_PERCTRL10 bit[17] to 0 to soft-reset the SATA PHY.

The SATA Host controller also provides the following two synchronous reset policies:

- When SATA_GHC_GHC bit[0] of the SATA Host controller is set to 1, the SATA Host switches to the reset mode. In this case, the logics of all internal modules are restored to initial values. After reset, SATA_GHC_GHC bit[0] is cleared automatically.
- When the value of SATA_PORT_CMD bit[0] corresponding to port0 or port1 is changed from 1 to 0, port 0 or port1 enters the reset mode.

9.5.4 Configuration of the Operating Mode

Before starting the SATA Host, you must initialize the SATA PHY to ensure proper running of the SATA PHY and perform initialization negotiation between the SATA Host and SATA Device.

Initializing the SATA PHY

By taking the 125 MHz on-chip reference clock and 1.5 Gbps port0 as examples, initialize the SATA PHY as follows:

Step 1 Set SC_PERCTRL9 bit[1] to 1 to select the 125 MHz CRG output reference clock.

Step 2 Set SC_PERCTRL10 bit[18] to 1 to cancel the soft reset on the SATA controller bus.

Step 3 Set SC_PERCTRL10 bit[19] to 1 to cancel the soft reset on the SATA PHY.

Step 4 Set SATA_PHY0_CTLH bit[12] to 1 to disable the reference clock provided to the PHY.

Step 5 Set SATA_PHY0_CTLL to 0x840E_C788.

Step 6 Set SATA_PHY0_CTLH to 0x2121.

Step 7 Set SATA_OOB_CTL to 0x8406_0C15.

Step 8 Set SATA_PORT_PHYCTL to 0x0E26_2709.

Step 9 Set SC_PERCTRL10 bit[19] to 0 to soft-reset the SATA PHY.

Step 10 Set SC_PERCTRL10 bit[19] to 1 to cancel the soft reset on the SATA PHY.

Step 11 Set SC_PERCTRL10 bit[17] to 1 to cancel the soft reset on the SATA controller interface.

Step 12 Set SC_PERCTRL10 bit[20] to 1 to cancel the soft reset on the clock domain tx0 of the SATA controller.

Step 13 Set SC_PERCTRL10 bit[22] to 1 to cancel the soft reset on the clock domain rx0 of the SATA controller.

Step 14 Set SC_PERCTRL10 bit[24] to 1 to cancel the soft reset on the clock domain alive of the SATA controller.

----End

Initialization Negotiation

After the PLL of the SATA PHY works properly, initialization negotiation is performed between the SATA Host and SATA Device as follows:

Step 1 Set SATA_PORT_CMD[cmd_sud] to 1.

Step 2 Wait until the indication signal phyrdy from the SATA PHY is valid. Then check whether SATA_PORT_SSTS[pxssts_det] is set to 3. If yes, it indicates that the corresponding port works properly and initialization is successful.

----End

Running Services

After initialization negotiation, run services as follows:

Step 1 Clear interrupts (skip this step after reset or if services are started initially). To be specific, set the values of SATA_PORT_SERR, SATA_PORT_IS, and SATA_GHC_IS to 0xFFFFFFFF.

Step 2 Configure the interrupt mask register SATA_PORT_IE to mask the interrupts that do not need to be reported.

Step 3 Enable the global interrupt by setting SATA_GHC_GHC to 0x80000002.

Step 4 Set up a linked list according to the description in section 9.8 "Appendix A Formats of the SATA Command Lists."

Step 5 Set the base address of the port command lists in the memory by configuring SATA_PORT_CLB[port_clb] and notify the transmit DMAC of the position for storing the commands and data to be read. The configured base address is the memory base address allocated for the port command lists.

Step 6 Set the base address in the memory for storing the frames received through the port by configuring SATA_PORT_FB[port_fb] and notify the receive DMAC of the position for storing the received frame information structures (FISs). The configured base address is the memory base address allocated for the port receive frames.

Step 7 Set SATA_PORT_CMD[st] to 1 to enable the transmit DMAC to transmit commands and data; set SATA_PORT_CMD[fre] to 1 to enable the receive DMAC to receive FISs and write them to the system memory.

Step 8 Configure the port command transmit register SATA_PORT_CI to indicate the command to be transmitted.

Step 9 Transmit the command and data.

Step 10 Check whether the current command is complete through the interrupt bit and command execution status. When interrupts are received, check whether all the CI bits are cleared for the PIO or DMA operation and check the CI and SACT bits are cleared for the NCQ operation.

Step 11 Repeat Step 1 to Step 10 for the next transfer if necessary.

----End

NOTE

- Perform the operations of legacy DMA, PIO, and ATAPI according to the preceding steps. The linked lists (such as the command codes and flag bits), however, are different for theses operations.
- For the NCQ operation, besides a different linked list, SATA_PORT_SACT also needs to be configured to indicate the number of executed commands during the NCQ operation. To be specific, configure SATA_PORT_SACT after Step 7 and ensure that the position of the commands configured by SATA_PORT_SACT map to those of the commands in SATA_PORT_CI.

9.6 Register Summary

Table 9-2 lists the value ranges and meanings of the variables in the offset addresses of registers.

Table 9-2 Variables in the offset address of registers

Variable	Value Range	Description
n	0 or 1	Indicates the two ports of a controller

9.6.1 Summary of the SATA Registers

Table 9-3 lists the SATA registers.

Table 9-3 Summary of the SATA registers (base address: 0x5202_0000)

Offset Address	Register	Description	Page
0x0000	SATA_GHC_CAP1	Feature support register 1	9-10
0x0004	SATA_GHC_GHC	Global control register	9-12
0x0008	SATA_GHC_IS	Interrupt status register	9-12
0x000C	SATA_GHC_PI	Port implementation register	9-13
0x0010	SATA_GHC_VS	AHCI version identifier register	9-14
0x0014	SATA_GHC_CCC_CTL	Command completion coalescing (CCC) control register	9-14
0x0018	SATA_GHC_CCC_PORTS	CCC port enable register	9-15
0x0024	SATA_GHC_CAP2	Feature support register 2	9-16
0x0028	SATA_GHC_BOHC	Basic input/output system (BIOS)/operating system (OS) handoff control register	9-16
0x0050	SATA_GHC_TM	TM test status register	9-17
0x0054	SATA_PHY0_CTLL	PHY0 global control register for lower bits	9-18
0x0058	SATA_PHY0_CTLH	PHY0 global control register for upper bits	9-19
0x005C	SATA_PHY0_STS	PHY0 global status register	9-21
0x0060–0x0068	RESERVED	Reserved	–
0x006C	SATA_OOB_CTL	PHY out of band (OOB) control register	9-24

9.6.2 Summary of the SATA_PORT_CFG Registers

Table 9-4 lists the SATA_PORT_CFG registers.

Table 9-4 Summary of the SATA_PORT_CFG registers (base address: 0x5202_0100)

Offset Address	Register	Description	Page
0x000+n×0x80	SATA_PORT_CLB	Command list base address register	9-24
0x008+n×0x80	SATA_PORT_FB	Receive FIS base address register	9-25
0x010+n×0x80	SATA_PORT_IS	Port interrupt status register	9-25
0x014+n×0x80	SATA_PORT_IE	Port interrupt mask register	9-27
0x018+n×0x80	SATA_PORT_CMD	Port command and status register	9-29
0x20+n×0x80	SATA_PORT_TFD	Port task file register	9-31
0x24+n×0x80	SATA_PORT_SIG	Port signature register	9-32
0x028+n×0x80	SATA_PORT_SSTS	Port status register	9-32
0x02C+n×0x80	SATA_PORT_SCTL	Port control register	9-33
0x30+n×0x80	SATA_PORT_SERR	Error diagnosis status register	9-34
0x034+n×0x80	SATA_PORT_SACT	NCQ command identifier control register	9-36
0x38+n×0x80	SATA_PORT_CI	Command transmit control register	9-36
0x3C+n×0x80	SATA_PORT_SNTF	Async notification event indication register	9-37
0x044+n×0x80	SATA_PORT_FIFOTH	Receive first in first out (FIFO) threshold register	9-37
0x050+n×0x80	SATA_PORT_HBA	Host bus adapter (HBA) test status register	9-38
0x054+n×0x80	SATA_PORT_LINK	Link test status register	9-39

Offset Address	Register	Description	Page
0x058+n×0x80	SATA_PORT_DMA1	DMAC test status register 1	9-40
0x05C+n×0x80	SATA_PORT_DMA2	DMAC test status register 2	9-41
0x060+n×0x80	SATA_PORT_DMA3	DMAC test status register 3	9-41
0x064+n×0x80	SATA_PORT_DMA4	DMAC test status register 4	9-41
0x068+n×0x80	SATAf_PORT_DMA5	DMAC test status register 5	9-42
0x6C+n×0x80	SATA_PORT_DMA6	DMAC test status register 6	9-43
0x070+n×0x80	SATA_PORT_DMA7	DMAC test status register 7	9-43
0x074+n×0x80	SATA_PORT_PHYCTL	PHY control register	9-44
0x078+n×0x80	SATA_PORT_PHYSTS	PHY test status register	9-46

9.7 Register Description

9.7.1 Description of the SATA Registers

SATA_GHC_CAP1

SATA_GHC_CAP1 is feature support register 1.

Offset Address	Register Name	Total Reset Value
0x0000	SATA_GHC_CAP1	0x6F26_FFA3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	s64a	sncq	ssntf	smps	sss	salp	sal	sclo	iss				reserved	sam	spm	fbss	pmd	ssc	psc	ncs					cccs	ems	sxs	np				
Reset	0	1	1	0	1	1	1	1	0	0	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1

Bits	Access	Name	Description
[31]	RO	s64a	Fixed at 0, it indicates that the 64-bit data structure cannot be accessed.

[30]	RO	sncq	Fixed at 1, it indicates that NCQ is supported.
[29]	RO	ssntf	Fixed at 1, it indicates that the port serial ATA notification (SNTF) register is supported.
[28]	RO	smps	Fixed at 0, it indicates that mechanical hot-plugging is not supported.
[27]	RO	sss	Fixed at 1, it indicates that staggered spin-up is supported.
[26]	RO	salp	Fixed at 1, it indicates that power supply management is supported.
[25]	RO	sal	Fixed at 1, it indicates that the LED pin is supported.
[24]	RO	sclo	Fixed at 1, it indicates that command list override is supported.
[23:20]	RO	iss	Fixed at 0x2, it indicates that the maximum rate of 3 Gbps is supported.
[19]	RO	reserved	Reserved.
[18]	RO	sam	Fixed at 1, it indicates that only the AHCI mode is supported.
[17]	RO	spm	Fixed at 1, it indicates that the port multiplier is supported.
[16]	RO	fbss	Fixed at 0, it indicates that FIS-based switching is not supported.
[15]	RO	pmd	Fixed at 1, it indicates that multiple DRQ blocks cannot be transferred in PIO mode.
[14]	RO	ssc	Fixed at 1, it indicates that the transition to the slumber state is supported.
[13]	RO	psc	Fixed at 1, it indicates that the transition to the partial state is supported.
[12:8]	RO	ncs	Fixed at 0x1F, it indicates that 32 command slots are supported.
[7]	RO	cccs	Fixed at 1, it indicates that the CCC function is supported.
[6]	RO	ems	Fixed at 0, it indicates that enclose management is not supported.
[5]	RO	sxs	Fixed at 1, it indicates that the external SATA interface is supported.
[4:0]	RO	np	Fixed at 0x03, it indicates that up to two ports are supported.

SATA_GHC_GHC

SATA_GHC_GHC is the global control register.

Offset Address	Register Name	Total Reset Value
0x0004	SATA_GHC_GHC	0x8000_0000

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

Bits	Access	Name	Description
[31]	RO	ahci_en	Fixed at 1, it indicates that the software can interact with the controller through the AHCI mechanism only.
[30:2]	RO	reserved	Reserved.
[1]	RW	int_enable	Controller interrupt enable. 0: disabled 1: enabled
[0]	RW	hba_rst	Soft reset control for the controller. 0: not reset 1: reset Writing 1 resets the controller and this bit is cleared after reset; writing 0 has no effect. In addition, reset has no effect on the registers SATA_GHC_BOHC, SATA_PORT_FB, and SATA_PORT_CLB.

SATA_GHC_IS

SATA_GHC_IS is the interrupt status register.

Offset Address	Register Name	Total Reset Value
0x0008	SATA_GHC_IS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ips_ccc	reserved																													ips_port1	ips_port0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	WC	ips_ccc	CCC interrupt status. 0: No CCC interrupt is generated. 1: A CCC interrupt is generated.

[30:2]	RO	reserved	Reserved.
[1]	WC	ips_port1	Interrupt status of port 1. 0: No interrupt is reported. 1: An interrupt is reported.
[0]	WC	ips_port0	Interrupt status of port 0. 0: No interrupt is reported. 1: An interrupt is reported.

SATA_GHC_PI

SATA_GHC_PI is the port implementation register.

Offset Address	Register Name	Total Reset Value
0x000C	SATA_GHC_PI	0x0000_000F

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1:0]	RO	port_imp	Port validity indication. bit[1] maps to port 1 and bit[0] maps to port 0. 0: The ports are invalid. 1: The ports are valid.

SATA_GHC_VS

SATA_GHC_VS is the AHCI version identifier register.

Offset Address	Register Name	Total Reset Value
0x0010	SATA_GHC_VS	0x0001_0200

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

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

Bits	Access	Name	Description
[31:0]	RO	ahci_vs	The supported AHCI version is V1.2.

SATA_GHC_CCC_CTL

SATA_GHC_CCC_CTL is the CCC control register.

Offset Address	Register Name	Total Reset Value
0x0014	SATA_GHC_CCC_CTL	0x0001_01F8

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ccc_tv																ccc_cc								ccc_int					reserved		ccc_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0

Bits	Access	Name	Description
[31:16]	RW	ccc_tv	CCC timeout parameter, in the unit of ms. When the CCC function is enabled, the timeout counter loads this parameter value. If a command is executed on a port involved in CCC counting, the counter decreases by 1 every 1 ms until a CCC interrupt is generated when the counter reaches 0. Then the counter reloads the parameter value for the next counting. Note that these bits cannot be set to 0s.
[15:8]	RW	ccc_cc	CCC command completion upper threshold. When the CCC function is enabled, the counter is cleared after commands are executed. Then the counter starts to count the number of completed commands on the ports involved in CCC counting. If the count value is equal to or greater than the parameter value, a CCC interrupt is generated. In this case, the counter is cleared again for the next counting. If the value 0 is written, the command completion interrupt is disabled and the CCC interrupt is generated in case of timeout only.
[7:3]	RO	ccc_int	CCC interrupt vector ID. If the value is 0x1F (31), SATA_GHC_IS bit[31] indicates the CCC interrupt status.
[2:1]	RO	reserved	Reserved.
[0]	RW	ccc_en	CCC function enable. 0: disabled 1: enabled When the CCC function is enabled, the values of ccc_tv and ccc_cc cannot be changed.

SATA_GHC_CCC_PORTS

SATA_GHC_CCC_PORTS is the CCC port enable register.

Offset Address	Register Name	Total Reset Value
0x0018	SATA_GHC_CCC_PORTS	0x0000_0000

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

Bits	Access	Name	Description
[31:2]	RO	reserved	Reserved.
[1:0]	RW	ccc_prt	Specifies the port that is involved in CCC counting. bit[1] maps to port 1 and bit[0] maps to port 0. 1: The port is involved in CCC counting. 0: The port is not involved in CCC counting. The value of this register can be changed at any time and the changed value takes effect immediately.

SATA_GHC_CAP2

SATA_GHC_CAP2 is feature support register 2.

Offset Address	Register Name	Total Reset Value
0x0024	SATA_GHC_CAP2	0x0000_0001

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RO	cap_boh	Fixed at 1, it indicates that BIOS/OS handoff control is supported.

SATA_GHC_BOHC

SATA_GHC_BOHC is the BIOS/OS handoff control register.

Offset Address	Register Name	Total Reset Value
0x0028	SATA_GHC_BOHC	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																											bohc_bb	bohc_ooc	bohc_sooe	bohc_oos	bohc_bos
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:5]	RO	reserved	Reserved.
[4]	RW	bohc_bb	BIOS status indication. 0: The BIOS is not busy. 1: The BIOS is busy performing certain operations and is ready to hand the control rights off to the OS.
[3]	WC	bohc_ooc	When the value of bohc_oos is changed from 0 to 1, this bit is fixed at 1. Writing 1 clears this bit and writing 0 has no effect.
[2]	RW	bohc_sooe	Message interrupt enable. 0: No message interrupt is generated. 1: When bohc_ooc is set to 1, a message interrupt is generated.
[1]	RW	bohc_oos	Request applied by the OS for controlling the controller. 0: The OS does not apply for the control rights to the controller. 1: The OS applies for the control rights to the controller. If bohc_oos is 1 and bios_bos is 0, it indicates that the OS has obtained the control rights to the SATA controller. Resetting the SATA controller through SATA_GHC_GHC[hab_rst] has no effect on this bit.
[0]	RW	bohc_bos	Flag that indicates that the BIOS has the control rights to the controller. 0: The BIOS does not have the control rights to the controller. 1: The BIOS has the control rights to the controller. If the OS applies for the controller right of the controller, the BIOS clears this bit. Resetting the SATA controller through SATA_GHC_GHC[hab_rst] has no effect on this bit.

SATA_GHC_TM

SATA_GHC_TM is the TM test status register.

Offset Address	Register Name	Total Reset Value
0x0050	SATA_GHC_TM	0x0000_0000

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

Bits	Access	Name	Description
[31:3]	RO	reserved	Reserved.
[2:0]	RO	req_sel	Current DMAC that has the right to use the AHB master. 0x0: transmit DMAC of port 0 0x1: receive DMAC of port 0 0x2: transmit DMAC of port 1 0x3: receive DMAC of port 1 Others: reserved

SATA_PHY0_CTLL

SATA_PHY0_CTLL is the PHY0 global control register for lower bits.

Offset Address	Register Name	Total Reset Value
0x0054	SATA_PHY0_CTLL	0x8D0E_C88A

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	mpll_prescale		mpll_ncy					mpll_ncy5	mpll_int_ctl				mpll_prop_ctl				tx_lvl				los_lvl				acjt_lvl				reserved		pddq_h	
Reset	1	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	1	1	0	0	1	0	0	0	1	0	0	0	1	0	1	0

Bits	Access	Name	Description
[31:30]	RW	mpll_prescale	When the reference clock is changed, this value also needs to be changed. 00: ref_clk is used as it is. 01: ref_clk is multiplied by 2. 10: ref_clk is divided by 2 11: Reserved.
[29:25]	RW	mpll_ncy	Operating parameter of the internal MPLL of the PHY. These bits need to be used with mpll_ncy5 and indicate the used multiplier.

[24:23]	RW	mpll_ncy5	Operating parameter of the internal MPLL of the PHY. These bits need to be used with mpll_ncy and indicate the used multiplier.
[22:20]	RW	mpll_int_ctl	Internal bandwidth control and select signal of the MPLL. These bits must be set to 0b000.
[19:17]	RW	mpll_prop_ctl	Internal proportional bandwidth control signal of the MPLL. These bits must be set to 0b111 and can be written only when the register is reset or the MPLL is invalid.
[16:12]	RW	tx_lvl	Transmit level parameter that is relevant to the selected SATA protocol. These bits must be set to 0b01100.
[11:7]	RW	los_lvl	Loss of signal (LOS) detection level control signal. When the rate of the SATA port is 1.5 Gbps, set these bits to 0b01111. When the rate of the SATA port is 3 Gbps, set these bits to 0b10001.
[6:2]	RW	acjt_lvl	ACJTAG receiver comparator level control signal. These bits must be set to 0b00010.
[1]	RO	reserved	Reserved.
[0]	RW	pddq_h	IDDQ test signal. To perform an IDDQ test, all the lanes and support blocks must be powered off before pddq_h is valid. In normal mode, this bit must be set to 0.

SATA_PHY0_CTLH

SATA_PHY0_CTLH is the PHY0 global control register for upper bits.

Offset Address	Register Name	Total Reset Value
0x0058	SATA_PHY0_CTLH	0x0000_2125

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																		use_refclk_alt	mpll_ck_off	mpll_pwron	mpll_ss_en	cko_word_con				cko_alive_con		rtune_do_tune		reserved		reset_n	wide_xface
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	1

Bits	Access	Name	Description
[31:14]	RO	reserved	Reserved.
[13]	RW	use_refclk_alt	Select signal of the PHY reference clock. 0: refclk differential signals 1: refclk_alt differential signals

[12]	RW	mpll_ck_off	Power-on control signal of the MPLL. Control this signal by conforming to the following rules: 1 Before providing refclk to the MPLL, set mpll_ck_off to 0. 2 Before setting mpll_ck_off to 0, set mpll_ncy, mpll_ncy5, and, pll_prescale to proper values. 3 If refclk is paused or switched or the values of mpll_ncy, mpll_ncy5, and mpll_prescale need to be changed, set mpll_ck_off to 1 first.
[11]	RW	mpll_pwron	Power-on of the MPLL. 0: The cko_word clock is invalid. 1: The internal MPLL is reset and the cko_word clock is generated based on the frequency of refclk. Before the MPLL is disabled, tx_en must be OFF or in the CM state and rx_en and rx_pll_pwron must be set to 0.
[10]	RW	mpll_ss_en	Spread spectrum enable signal. 0: disabled 1: enabled If refclk is processed through the spread spectrum technology, this bit must be set to 0.
[9:7]	RW	cko_word_con	cko_word output select signal.
[6:5]	RW	cko_alive_con	cko_alive output select signal. 0: Invalid. 01: Remain the output frequency of the prescaler. 10: Low-frequency output, that is, 1/16 of the prescaler. 11: Reserved.
[4]	RW	rtune_do_tune	Resistor tune enable signal. 0: Do not tune the resistor. 1: Re-tune the resistor.
[3:2]	RW	reserved	Reserved.
[1]	RW	reset_n	Reset signal. This signal must be remained for 5 ns at least.
[0]	RW	wide_xface	Interface bit width control signal. 0: 10 bits 1: 20 bits

SATA_PHY0_STS

SATA_PHY0_STS is the PHY0 global status register.

Offset Address	Register Name	Total Reset Value
0x005C	SATA_PHY0_STS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	phy0_sts																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:0]	RO	phy0_sts	SATA PHY0 common status register.

SATA_OOB_CTL

SATA_OOB_CTL is the PHY OOB control register.

Offset Address	Register Name	Total Reset Value
0x006C	SATA_OOB_CTL	0x8406_0C15

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	oob_ctrl_valid	min_comiwake							max_comwake								min_cominit								max_cominit							
Reset	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	1

Bits	Access	Name	Description
[31]	RW	oob_ctrl_valid	Configuration bit of the OOB detection parameter. For high level, this bit is used to select the parameter of this register to be configured.
[30:24]	RW	min_comiwake	Minimum space required for the COMWAKE space detection.
[23:16]	RW	max_comwake	Maximum space required for the COMWAKE space detection.
[15:8]	RW	min_cominit	Minimum space required for the COMINIT space detection.
[7:0]	RW	max_cominit	Maximum space required for the COMINIT space detection.

9.7.2 Description of the SATA_PORT_CFG Registers

SATA_PORT_CLB

SATA_PORT_CLB is the command list base address register.

Offset Address	Register Name	Total Reset Value
0x000+nx0x80	SATA_PORT_CLB	0x0000_0000

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

Bits	Access	Name	Description
[31:10]	RW	port_clb	Base address for storing the port command list in the memory. Resetting the SATA controller through SATA_GHC_GHC[hab_rst] has no effect on these bits.
[9:0]	RO	reserved	Reserved.

SATA_PORT_FB

SATA_PORT_FB is the receive FIS base address register.

Offset Address	Register Name	Total Reset Value
0x008+nx0x80	SATA_PORT_FB	0x0000_0000

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

Bits	Access	Name	Description
[31:8]	RW	port_fb	Base address in the memory for storing the frames received through the port. Resetting the SATA controller through SATA_GHC_GHC[hab_rst] has no effect on these bits.
[7:0]	RO	reserved	Reserved.

SATA_PORT_IS

SATA_PORT_IS is the port interrupt status register.

Offset Address	Register Name	Total Reset Value
0x010+nx0x80	SATA_PORT_IS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	pxis_tfes	reserved	pxis_hbds	pxis_ifs	pxis_infs	reserved	pxis_ofs	pxis_ipms	pxis_prcs	reserved														pxis_pcs	pxis_dps	pxis_ufs	pxis_sdbs	pxis_dss	pxis_pss	pxis_drhs

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RO	reserved	Reserved.
[30]	WC	pxis_tfes	Task file data (TFD) error interrupt status. 0: The value of SATA_PORT_TFD[tfd_sts] ERR is not 1. 1: The value of SATA_PORT_TFD[tfd_sts] ERR is 1.
[29]	RO	reserved	Reserved.
[28]	WC	pxis_hbds	Internal bus error interrupt. 0: The DMAC accesses the memory properly. 1: An error occurs when the DMAC accesses the memory.
[27]	WC	pxis_ifs	Fatal error interrupt status. 0: No error occurs during the data frame transfer. 1: An error occurs during the data frame transfer.
[26]	WC	pxis_infs	Non-fatal error interrupt status. 0: No error occurs during the non-data frame transfer. 1: An error occurs during the non-data frame transfer.
[25]	RO	reserved	Reserved.
[24]	WC	pxis_ofs	Data transfer overflow interrupt status. 0: No overflow is detected. 1: During the data frame transfer, if the size of the data memory occupied by commands is smaller than the actual data amount, an interrupt is reported at the end of the data transfer.
[23]	WC	pxis_ipms	PM port number error interrupt status. 0: No PM port number error is detected during data receive. 1: A PM port number error is detected during data receive.
[22]	RO	pxis_prcs	PHY state change interrupt status. 0: No changes of the phyrdy signals are detected. 1: Changes of the phyrdy signals are detected. This bit directly reflects the value of SATA_PORT_SERR[diag_n].
[21:7]	RO	reserved	Reserved.
[6]	RO	pxis_pcs	Port connection change interrupt status. 0: No COMINIT signal transmitted from the device is detected. 1: A COMINIT signal transmitted from the device is detected. This bit directly reflects the value of SATA_PORT_SERR[diag.x].

[5]	WC	pxis_dps	Linked list transfer completion interrupt status. 0: The transfer of the linked list data is complete when no I bit in the PRD is 1. 1: The transfer of the linked list data is complete when an I bit in the PRD is 1.
[4]	RO	pxis_ufs	Unknown FIS interrupt status. 0: No unknown FIS is received. 1: An unknown FIS is received.
[3]	WC	pxis_sdbs	Set device bits FIS interrupt status. 0: No effect. 1: A set device bits FIS is received and the I bit is 1.
[2]	WC	pxis_dss	DMA setup FIS interrupt status. 0: No effect. 1: A DMA setup FIS is received and the I bit is 1.
[1]	WC	pxis_pss	PIO setup FIS interrupt status. 0: No effect. 1: A PIO setup FIS is received and the I bit is 1.
[0]	WC	pxis_drhs	D2H register FIS interrupt status. 0: No effect. 1: A D2H register FIS is received and the I bit is 1.

SATA_PORT_IE

SATA_PORT_IE is the port interrupt mask register.

Offset Address	Register Name	Total Reset Value
0x014+nx0x80	SATA_PORT_IE	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	pxie_tfee	reserved	pxie_hbde	pxie_ife	pxie_infe	reserved	pxie_ofe	pxie_ipme	pxie_prce	reserved														pxie_pce	pxie_dpe	pxie_ufe	pxie_sdbe	pxie_dse	pxie_pse	pxie_drhe	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RO	reserved	Reserved.
[30]	RW	pxie_tfee	TFD error interrupt mask. 0: masked 1: not masked

[29]	RO	reserved	Reserved.
[28]	RW	pxie_hbde	Internal bus error interrupt mask. 0: masked 1: not masked
[27]	RW	pxie_ife	Fatal error interrupt mask. 0: masked 1: not masked
[26]	RW	pxie_infe	Non-fatal error interrupt mask. 0: masked 1: not masked
[25]	RO	reserved	Reserved.
[24]	RW	pxie_ofe	Data transfer overflow interrupt mask. 0: masked 1: not masked
[23]	RW	pxie_ipme	PM port error interrupt mask. 0: masked 1: not masked
[22]	RW	pxie_prce	PHY state change interrupt mask. 0: masked 1: not masked
[21:7]	RO	reserved	Reserved.
[6]	RW	pxie_pce	Port connection change interrupt mask. 0: masked 1: not masked
[5]	RW	pxie_dpe	Linked list transfer completion interrupt mask. 0: masked 1: not masked
[4]	RW	pxie_ufe	Unknown FIS interrupt mask. 0: masked 1: not masked
[3]	RW	pxie_sdbe	Set device bits FIS interrupt mask. 0: masked 1: not masked
[2]	RW	pxie_dse	DMA setup FIS interrupt mask. 0: masked 1: not masked

[1]	RW	pxie_pse	PIO setup FIS interrupt mask. 0: masked 1: not masked
[0]	RW	pxie_drhe	D2H register FIS interrupt mask. 0: masked 1: not masked

SATA_PORT_CMD

SATA_PORT_CMD is the port command and status register.

Offset Address	Register Name	Total Reset Value
0x018+nx0x80	SATA_PORT_CMD	0x0020_0004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	cmd_icc				cmd_asp	cmd_alpe	cmd_dlae	cmd_atapi	reserved		cmd_esp	reserved			cmd_pma	reserved	cmd_cr	cmd_fr	reserved	cmd_ccs					reserved			cmd_fre	cmd_clo	reserved	cmd_sud	cmd_st
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits	Access	Name	Description
[31:28]	RW	cmd_icc	Port communications control signal. 0x0: No operation. It indicates that the next port state request is allowed. 0x1: Request to enable the port to be in the active state. 0x2: Request to enable the port to be in the partial state. 0x6: Request to enable the port to be in the slumber state. Others: Reserved. When the software writes any of the preceding values rather than the reserved values, the controller clears the cmd_icc bit after performing related operations. When the software requests the current state of the port, the controller clears the cmd_icc bit directly. If the software requests the port state change from a low-power state to another lower-power state, such as from the partial state to the slumber state, it needs to request the state change from the partial state, the active state, and then to the slumber state.
[27]	RW	cmd_asp	Slumber or partial state select for power management. 0: Actively enter the partial state. 1: Actively enter the slumber state.

[26]	RW	cmd_alpe	Automatic power management enable. 0: Disabled 1: Enabled. If SATA_PORT_CI and SATA_PORT_SACT are cleared, the controller enters the power-management state automatically. To be specific, if cmd_asp is 1, the controller enters the partial state; if cmd_asp is 0, the controller enters the slumber state.
[25]	RW	cmd_dlae	LED drive enable in ATAPI mode. 0: The LED pin can be driven when cmd_atapi is 0 and commands are being executed. 1: The LED pin can be driven when commands are being executed.
[24]	RW	cmd_atapi	ATAPI device indication. 0: The current device is not an ATAPI device. 1: The current device is an ATAPI device.
[23:22]	RO	reserved	Reserved.
[21]	RO	cmd_esp	Fixed at 1, it indicates that the external SATA device is supported.
[20:18]	RO	reserved	Reserved.
[17]	RW	cmd_pma	Port multiplier detection indication. 0: No port multiplier is connected to the port. 1: A port multiplier is connected to the port.
[16]	RO	reserved	Reserved.
[15]	RO	cmd_cr	Command list processing indication. 0: No command is being executed. 1: A command is being executed.
[14]	RO	cmd_fr	FIS receive processing indication. 0: No FIS is being received. 1: An FIS is being received.
[13]	RO	reserved	Reserved.
[12:8]	RO	cmd_ccs	Slot number of the current command. These bits are valid when cmd_st is 1 and are cleared when cmd_st is 0.
[7:5]	RO	reserved	Reserved.
[4]	RW	cmd_fre	FIS receive enable control. 0: Forbid to write the received FISs to the system memory. 1: Enable the received FISs and write them to the system memory. The software needs to set the receive FIS base address register SATA_PORT_FB before enabling this bit to receive FISs. When cmd_st is 1, this bit must be set to 1.

[3]	RW	cmd_clo	BSY/DQR clear control. The software can forcibly clear the BSY and DRQ bits through the cmd_clo bit and transmit commands to the device. 0: No effect. 1: Clear the BSY and DRQ bits of SATA_PORT_TFD[tfd_sts]. After the BSY and DRQ bits are 0, the cmd_clo bit is cleared automatically. The cmd_clo bit can be written as 1 only before the value of cmd_st is changed from 0 to 1. In addition, the software must write cmd_st as 1 after the cmd_clo bit is cleared.
[2]	RO	reserved	Reserved.
[1]	RW	cmd_sud	Spin-up device control. 0: When SATA_PORT_SCTL[det] is 0, the controller enters the listen mode. 1: After the system is powered on or the HBA is reset, the controller is enabled to transmit a COMRESET sequence to initialize the hardware device.
[0]	RW	cmd_st	Command list processing enable. 0: The controller becomes idle. 1: The controller processes the commands from slot 0 that are identified as valid slots by SATA_PORT_CI. **Note:** The cmd_st bit can be set to 1 only after cmd_fre is 1.

SATA_PORT_TFD

SATA_PORT_TFD is the port task file register.

Offset Address	Register Name	Total Reset Value
0x20+nx0x80	SATA_PORT_TFD	0x0000_007F

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:8]	RO	tfd_err	Task file error register value. The controller updates these bits after receiving a D2H register FIS, a PIO setup FIS, or a set device bits (SDB) FIS.

[7:0]	RO	tfd_sts	Task file status register value. bit[7]: BSY bit. It indicates that the device is busy. bit[6:4]: The meaning of these bits varies according to commands. bit[3]: DRQ bit. It indicates that there is the data to be transferred in the device. bit[2:1]: The meaning of these bits varies according to commands. bit[0]: ERR bit. It indicates that an error occurs during the data transfer. The controller updates these bits after receiving a D2H register FIS, a PIO setup FIS, or an SDB FIS.

SATA_PORT_SIG

SATA_PORT_SIG is the port signature register.

Offset Address	Register Name	Total Reset Value
0x24+nx0x80	SATA_PORT_SIG	0xFFFF_FFFF

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

Bits	Access	Name	Description
[31:0]	RO	signature	LBA address and sector addressing. The allocated addresses are as follows: bit[31:24]: LBA upper-bit address bit[23:16]: LBA middle address bit[15:8]: LBA lower-bit address bit[7:0]: number of sectors The controller updates this register when receiving the first D2H register FIS after the hardware device is reset.

SATA_PORT_SSTS

SATA_PORT_SSTS is the port status register.

Offset Address	Register Name	Total Reset Value
0x028+nx0x80	SATA_PORT_SSTS	0x0000_0100

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

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:12]	RO	reserved	Reserved.
[11:8]	RO	pxssts_ipm	Current port status. 0x0: No devices or no communications are set up. 0x1: Active. 0x2: Partial. 0x6: Slumber. Others: Reserved.
[7:4]	RO	pxssts_spd	Port negotiation speed status. 0x0: No devices or no communications are set up. 0x1: Negotiation to rate 1 for communications. 0x2: Negotiation to rate 2 for communications. 0x3: Negotiation to rate 3 for communications. Others: Reserved.
[3:0]	RO	pxssts_det	Device detection and PHY status. 0x0: No device is detected and no PHY communications are set up. 0x0: A device is detected but no PHY communications are set up. 0x0: A device is detected but the PHY communications are set up. 0x4: The PHY is offline or in the built-in self test (BIST) state. Others: Reserved.

SATA_PORT_SCTL

SATA_PORT_SCTL is the port control register.

Offset Address	Register Name	Total Reset Value
0x02C+nx0x80	SATA_PORT_SCTL	0x0000_0000

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

Bits	Access	Name	Description
[31:12]	RO	reserved	Reserved.

[11:8]	RW	pxsctl_ipm	Port power management status control. 0x0: No requirements. 0x1: Forbid to enter the partial state. 0x2: Forbid to enter the slumber state. 0x3: Forbid to enter the partial or slumber state. Others: Reserved.
[7:4]	RW	pxsctl_spd	Port communications speed control. 0x0: No requirements. 0x1: Limit the communications speed to rate 1. 0x2: Limit the communications speed to rate 2. 0x3: Limit the communications speed to rate 3. Others: Reserved.
[3:0]	RW	pxsctl_det	Device detection and port initialization control. 0x0: No device detection or initialization request. 0x1: Request the port to reset the initialization sequence COMRESET. 0x4: Force the port to be offline. Others: Reserved. When pxsctl_det is set to 1, the controller transmits the COMRESET sequence to the device. In this case, to ensure that the device receives the COMRESET sequence, the software must remain the value of pxsctl_det for at least 1 ms.

SATA_PORT_SERR

SATA_PORT_SERR is the error diagnosis status register.

Offset Address	Register Name	Total Reset Value
0x30+nx0x80	SATA_PORT_SERR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					diag_x	diag_f	reserved	diag_s	diag_h	diag_c	reserved	diag_b	diag_w	diag_i	diag_n	reserved					err_p	reserved	err_t	reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:27]	RO	reserved	Reserved.
[26]	WC	diag_x	Device detection status. 0: No COMINIT signal transmitted from the device is detected. 1: A COMINIT signal transmitted from the device is detected.

[25]	WC	diag_f	Detection status of the unknown FIS. 0: No unknown FIS is received. 1: An unknown FIS is received and this bit is set to 1 when the cyclic redundancy check (CRC) is correct.
[24]	RO	reserved	Reserved.
[23]	WC	diag_s	Link layer error status. 0: No state transition error occurs at the link layer. 1: A state transition error occurs at the link layer.
[22]	WC	diag_h	Handshake error status. 0: No R_ERR primitive transmitted from the device is received. 1: One or more R_ERR primitives transmitted from the device are received.
[21]	WC	diag_c	CRC error status. 0: No CRC error occurs during FIS receiving. 1: A CRC error occurs during FIS receiving.
[20]	RO	reserved	Reserved.
[19]	WC	diag_b	Decoding error status. 0: No 8b/10b decoding error is detected. 1: An 8b/10b decoding error is detected.
[18]	WC	diag_w	COMWAKE status. 0: No COMWAKE signal transmitted from the device is detected. 1: A COMWAKE signal transmitted from the device is detected.
[17]	WC	diag_i	PHY internal error status. 0: No PHY internal error is detected. 1: A PHY internal error is detected.
[16]	WC	diag_n	PhyRdy signal change status. 0: The PhyRdy signal is not changed. 1: The PhyRdy signal is changed. This bit is set to 1 when the value of the PhyRdy signal is changed from 1 to 0 or from 0 to 1.
[15:11]	RO	reserved	Reserved.
[10]	WC	err_p	SATA protocol incompliance error status. 0: No device behaviors do not comply with the SATA protocol. 1: Certain device behaviors do not comply with the SATA protocol.
[9]	RO	reserved	Reserved.

[8]	WC	err_t	Data integrity error status. 0: No data integrity error is detected. 1: A data integrity error is detected.
[7:0]	RO	reserved	Reserved.

SATA_PORT_SACT

SATA_PORT_SACT is the NCQ command identifier control register.

Offset Address	Register Name	Total Reset Value
0x034+nx0x80	SATA_PORT_SACT	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RW	port_sact	NCQ command identifier control. Each bit of this register maps to a tag ID and an NCQ command in the memory. To be specific, bit[31:0] map to the commands of slot 31–0 and tag 31–0 respectively. The following describes the meaning of each bit by taking bit[3] as an example: 0: The slot 3 command is a non-NCQ command. 1: The slot 3 command is an NCQ command. Before setting SATA_PORT_SACT bit[3] to 1, the software must clear SATA_PORT_CI bit[3]. After the command data transfer, the device transmits an SDB FIS. Then, the controller clears SATA_PORT_SACT bit[3] based on the SActive in the FIS. The software can set SATA_PORT_SACT only when cmd_st is 1. When cmd_st is 0, all bits of SATA_PORT_SACT are cleared.

SATA_PORT_CI

SATA_PORT_CI is the command transmit control register.

Offset Address	Register Name	Total Reset Value
0x38+nx0x80	SATA_PORT_CI	0x0000_0000

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

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

Bits	Access	Name	Description
[31:0]	RW	port_ci	Control for the commands to be transmitted. Each bit of this register maps to a command in the memory. To be specific, bit[31:0] map to the commands of slot 31–0 respectively. The following describes the meaning of each bit by taking bit[3] as an example: 0: There is no slot 3 command to be transmitted and executed. 1: A slot 3 command is created in the memory. Then the controller can transmit this command. After running this command and receiving a corresponding FIS, the controller clears SATA_PORT_CI bit[3] and the BSY, DRQ, and ERR bits of SATA_PORT_TFD. The bits of SATA_PORT_CI can be set to 1 only when cmd_st is 1 and all bits of SATA_PORT_CI are cleared when cmd_st is 0.

SATA_PORT_SNTF

SATA_PORT_SNTF is the async notification event indication register.

Offset Address	Register Name	Total Reset Value
0x3C+nx0x80	SATA_PORT_SNTF	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	WC	pxsntf_pmn	Async notification event status. If the controller receives an SDB FIS from the device on the PM port and the N bit of this FIS is 1, the controller sets the bit of this register corresponding to this port number to 1. The following describes the meaning of each bit by taking bit[3] as an example: 0: No async notification event occurs on the device whose PM port number is 3. 1: An async notification event occurs on the device whose PM port number is 3.

SATA_PORT_FIFOTH

SATA_PORT_FIFOTH is the receive FIFO threshold register.

Offset Address	Register Name	Total Reset Value
0x044+nx0x80	SATA_PORT_FIFOTH	0x0000_010C

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

Bits	Access	Name	Description
[31:9]	RW	reserved	Reserved.
[8:4]	RW	dmac_rxfifo_th	Flow control threshold of the DMAC receive FIFO. During data receiving, if the data amount in the DMAC FIFO is above the threshold, the controller starts to control the data flow.
[3]	RW	rxfifo_th_sel	Flow control FIFO select. 0: The flow control for the link receive FIFO is valid. 1: The flow control for the DMAC receive FIFO is valid.
[2:0]	RW	link_rxfifo_th	Flow control threshold of the link receive FIFO. During data receiving, if the data amount in the DMAC FIFO is above the threshold, the controller starts to control the data flow.

SATA_PORT_HBA

SATA_PORT_HBA is the HBA test status register.

Offset Address	Register Name	Total Reset Value
0x050+nx0x80	SATA_PORT_HBA	0x0100_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				p_curr_st				reserved				ndr_curr_st				cfis_curr_st				reserved	pio_curr_st			reserved	pm_curr_st			reserved	err_curr_st		
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:28]	RO	reserved	Reserved.
[27:24]	RO	p_curr_st	Current state of the HBA_PINIT_STATE state machine.
[23:21]	RO	reserved	Reserved.

[20:16]	RO	ndr_curr_st	Current state of the HBA_NDR_STATE state machine.
[15:12]	RO	cfis_curr_st	Current state of the HBA_CFIS_STATE state machine.
[11]	RO	reserved	Reserved.
[10:8]	RO	pio_curr_st	Current state of the HBA_PIO_STATE state machine.
[7]	RO	reserved	Reserved.
[6:4]	RO	pm_curr_st	Current state of the HBA_PM_STATE state machine.
[3]	RO	reserved	Reserved.
[2:0]	RO	err_curr_st	Current state of the HBA_ERR_STATE state machine.

SATA_PORT_LINK

SATA_PORT_LINK is the link test status register.

Offset Address	Register Name	Total Reset Value
0x054+nx0x80	SATA_PORT_LINK	0x0320_2020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				link_curr_st				reserved	link_df_fifo_full	link_df_fifo_empty	link_df_fifo_count					reserved	link_rx_fifo_full	link_rx_fifo_empty	link_rx_fifo_count					reserved	link_tx_fifo_full	link_tx_fifo_empty	link_tx_fifo_count				
Reset	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Access	Name	Description
[31:29]	RO	reserved	Reserved.
[28:24]	RO	link_curr_st	Current state of the LINK_CTL_STATE state machine.
[23]	RO	reserved	Reserved.
[22]	RO	link_df_fifo_full	Full flag of the link frequency difference FIFO. 0: not full 1: full
[21]	RO	link_df_fifo_empty	Empty flag of the link frequency difference FIFO. 0: not empty 1: empty
[20:16]	RO	link_df_fifo_count	Data amount in the link frequency difference FIFO.
[15]	RO	reserved	Reserved.

[14]	RO	link_rx_fifo_full	Full flag of the link receive FIFO. 0: not full 1: full
[13]	RO	link_rx_fifo_empty	Empty flag of the link receive FIFO. 0: not empty 1: empty
[12:8]	RO	link_rx_fifo_count	Data amount in the link receive FIFO.
[7]	RO	reserved	Reserved.
[6]	RO	link_tx_fifo_full	Full flag of the link transmit FIFO. 0: not full 1: full
[5]	RO	link_tx_fifo_empty	Empty flag of the link transmit FIFO. 0: not empty 1: empty
[4:0]	RO	link_tx_fifo_count	Data amount in the link transmit FIFO.

SATA_PORT_DMA1

SATA_PORT_DMA1 is DMAC test status register 1.

Offset Address	Register Name	Total Reset Value
0x058+nx0x80	SATA_PORT_DMA1	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	RO	reserved	Reserved.
[27:24]	RO	txdmac_cur_state	Current state of the SATA_TX_DMAC state machine.
[23]	RO	txdmac_prd_i	I bit in the entry of the PRD linked list of SATA_TX_DAMC.
[22:0]	RO	tx_entry_dbc_cnt	Down counter in SATA_TX_DMAC. It indicates the number of data bytes in the current entry.

SATA_PORT_DMA2

SATA_PORT_DMA2 is DMAC test status register 2.

Offset Address	Register Name	Total Reset Value
0x05C+nx0x80	SATA_PORT_DMA2	0x0020_0000

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

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:8]	RO	tx_data_fis_cnt	Down counter in SATA_TX_DMAC. It indicates the number of data FIS bytes during the operation of PIO, legacy DMA, or first party DMA. For the PIO operation, the initial value is equal to the value of transcount in the PIO setup FIS; for the legacy DMA or first party DMA operation, the initial value is 16'h2000 (2048 DWORD).
[7:0]	RO	tx_cmdh_prdtl	Down counter in SATA_TX_DMAC. The parameter in the command header indicates the number of entries in the physical region descriptor table (PRDT).

SATA_PORT_DMA3

SATA_PORT_DMA3 is DMAC test status register 3.

Offset Address	Register Name	Total Reset Value
0x060+nx0x80	SATA_PORT_DMA3	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	tx_fpdma_tran_cnt	Down counter in SATA_TX_DMAC. It indicates the number of data FIS bytes during the first party DMA operation. The initial value is equal to the value of transcount in the DMA setup FIS.

SATA_PORT_DMA4

SATA_PORT_DMA4 is DMAC test status register 4.

Offset Address	Register Name	Total Reset Value
0x064+nx0x80	SATA_PORT_DMA4	0x0000_0000

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

Bits	Access	Name	Description
[31:28]	RO	reserved	Reserved.
[27:24]	RO	rxdmac_cur_state	Current state of the SATA_RX_DMAC state machine.
[23]	RO	rxdmac_prd_i	I bit in the entry of the PRD linked list of SATA_RX_DAMC.
[22:0]	RO	rx_entry_dbc_cnt	Down counter in SATA_RX_DMAC. It indicates the number of data bytes in the current entry.

SATA_PORT_DMA5

SATA_PORT_DMA5 is DMAC test status register 5.

Offset Address	Register Name	Total Reset Value
0x068+nx0x80	SATA_PORT_DMA5	0x0020_0000

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

Bits	Access	Name	Description
[31:24]	RO	reserved	Reserved.
[23:8]	RO	rx_data_fis_cnt	Down counter in SATA_RX_DMAC. It indicates the number of data FIS bytes during the operation of PIO, legacy DMA, or first party DMA. For the PIO operation, the initial value is equal to the value of transcount in the PIO setup FIS; for the legacy DMA or first party DMA operation, the initial value is 0x2000 (2048 DWORD).
[7:0]	RO	rx_cmdh_prdtl	Down counter in SATA_RX_DMAC. The parameter in the command header indicates the number of entries in the PRDT.

SATA_PORT_DMA6

SATA_PORT_DMA6 is DMAC test status register 6.

Offset Address	Register Name	Total Reset Value
0x6C+nx0x80	SATA_PORT_DMA6	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	rx_fpdma_tran_cnt	Down counter in SATA_RX_DMAC. It indicates the number of data FIS bytes during the first party DMA operation. The initial value is equal to the value of transcount in the DMA setup FIS.

SATA_PORT_DMA7

SATA_PORT_DMA7 is DMAC test status register 7.

Offset Address	Register Name	Total Reset Value
0x070+nx0x80	SATA_PORT_DMA7	0x0005_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										pio_op	fpdma_op	dmac_rx_fifo_full	dmac_rx_fifo_empty	dmac_tx_fifo_full	dmac_tx_fifo_empty	dmac_rx_fifo_cnt								dmac_tx_fifo_cnt							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:22]	RO	reserved	Reserved.
[21]	RO	pio_op	PIO operation indication. 0: The current command is not used for the PIO operation. 1: The current command is for the PIO operation.
[20]	RO	fpdma_op	First party DMA operation indication. 0: The current command is not used for the first party DMA operation. 1: The current command is for the first party DMA operation.

[19]	RO	dmac_rx_fifo_full	Full status of SATA_DMAC_RX_FIFO. 0: not full 1: full
[18]	RO	dmac_rx_fifo_empty	Empty status of SATA_DMAC_RX_FIFO. 0: not empty 1: empty
[17]	RO	dmac_tx_fifo_full	Full status of SATA_DMAC_TX_FIFO. 0: not full 1: full
[16]	RO	dmac_tx_fifo_empty	Empty status of SATA_DMAC_TX_FIFO. 0: not empty 1: empty
[15:8]	RO	dmac_rx_fifo_cnt	Number of data segments in SATA_DMAC_RX_FIFO (in DWORD).
[7:0]	RO	dmac_tx_fifo_cnt	Number of data segments in SATA_DMAC_TX_FIFO (in DWORD).

SATA_PORT_PHYCTL

SATA_PORT_PHYCTL is the PHY control register.

Offset Address	Register Name	Total Reset Value
0x074+nx0x80	SATA_PORT_PHYCTL	0x0E63_6159

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved			phy_disable	phy_calibrated	spd_change_ack	dp_rdy	bist_tx_fspd	neg_mode_b	gen2_en	los_ctl		rx_dpll_mode				rx_eq_val				rx_term_en	tx_calc	tx_edgerate		tx_cko_en	rx_align_en	tx_clk_align	tx_atten				tx_boost
Reset	0	0	0	0	1	1	1	0	0	1	1	0	0	0	1	1	0	1	1	0	0	0	0	1	0	1	0	1	1	0	0	1

Bits	Access	Name	Description
[31:29]	RO	reserved	Reserved.
[28]	RW	phy_disable	Whether to use the PHY. 0: used 1: not used
[27]	RW	phy_calibrated	Whether to calibrate the PHY. 0: not calibrated 1: calibrated

[26]	RW	spd_change_ack	Whether the rate is allowed to switch. 0: not allowed 1: allowed
[25]	RW	dp_rdy	Whether the PHY is ready to transmit data. 0: not ready 1: ready
[24]	RW	bist_tx_fspd	Whether the clock frequency is forced to transmit in BIST mode. 0: not forced 1: forced
[23]	RW	neg_mode_b	Negotiation mode B select. 0: not supported 1: supported
[22]	RW	gen2_en	Transmit control signal. It indicates whether to support the 3G mode. 0: not supported 1: supported
[21:20]	RW	los_ctl	LOS detection control. 00: The LOS detection is disabled. 01: Reserved. 10: The OOB signal is being detected. 11: Reserved.
[19:17]	RW	rx_dpll_mode	Control mode of the receive DPLL. 000: indicates PHUG is 1 and FRUG is 1. 001: indicates PHUG is 2 and FRUG is 2. 010: indicates PHUG is 1 and FRUG is 4. 011: indicates PHUG is 2 and FRUG is 4. Others: Reserved.
[16:14]	RW	rx_eq_val	Receive balancing control. The internal balancing value is ~ (rx_eq_val + 1) x 0.5 dB.
[13]	RW	rx_term_en	Receive termination enable. 0: disabled 1: enabled
[12]	RW	tx_calc	Reserved, fixed at 0.
[11:10]	RW	tx_edgerate	Edge control for the transmitted signal. When the rate of the SATA port is 1.5 Gbps, set these bits to 0b01. When the rate of the SATA port is 3 Gbps, set these bits to 0b00.

[9]	RW	tx_cko_en	tx_cko_clk clock enable. 0: disabled 1: enabled
[8]	RW	rx_align_en	Received data alignment. 0: not supported 1: supported
[7]	RW	tx_clk_align	Transmit clock alignment. 0: not aligned 1: aligned
[6:4]	RW	tx_atten	Transmit attenuation control. 000: 16/16 001: 14/16 010: 12/16 011: 10/16 100: 9/16 101: 8/16 11X: reserved
[3:0]	RW	tx_boost	Transmit boost control. The value is –20log [1- (tx_boost[3:0] + 0.5)/32]dB.

SATA_PORT_PHYSTS

SATA_PORT_PHYSTS is the PHY test status register.

Offset Address	Register Name	Total Reset Value
0x078+nx0x80	SATA_PORT_PHYSTS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved													tx_cko_word	tx_rxpres	tx_done	spd_change	link_rdy	init_compl	pwr_state	rx_pll_pwron	rx_en	tx_en			mpll_pwron	phy_comwake	phy_cominit	half_rate	phyrdy	los	op_done
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:19]	RO	reserved	Reserved.
[18]	RW	tx_cko_word	Indicates the word receive clock of each lane, reserved currently.
[17]	RW	tx_rxpres	Indicates the receive detection, reserved currently.
[16]	RW	tx_done	Indicates that the operation of transmitting part of requests is complete, active high and reserved currently.

[15]	RW	spd_change	Indicates the rate change
[14]	RW	link_rdy	Indicates that sufficient D10.2 is transmitted in high level.
[13]	RW	init_compl	Indicates that non-align primitive is received in high level and initialization is complete.
[12]	RW	pwr_state	Indicates the low-power mode (partial or slumber state) when the bit is 1; indicates the active state in other cases.
[11]	RW	rx_pll_pwron	Indicates the power-on reset signal of the receive PLL.
[10]	RW	rx_en	Indicates the control status of the rx_en signal.
[9:7]	RW	tx_en	Indicates the control status of the tx_en signal.
[6]	RW	mpll_pwron	Indicates the power-on control signal of the MPLL.
[5]	RW	phy_comwake	Indicates that the PHY detects the COMWAKE signal, active high.
[4]	RW	phy_cominit	Indicates that the PHY detects the COMINIT signal, active high.
[3]	RW	half_rate	Indicates the rate of 1.5 Gbps when the bit is 1.
[2]	RW	phyrdy	Indicates that the PHY is initialized and then the PHY can communicate with the link layer.
[1]	RW	los	Indicates the loss of signal output, active high.
[0]	RW	op_done	Indicates that the operations requested by the MPLL are complete, active high.

9.8 Appendix A Formats of the SATA Command Lists

Figure 9-5 shows the structure of the FIS linked list. The linked list refers to the areas that are created in the system memory by the software. The base address of the linked list is stored in the PxFB and PxFBU registers in the AHCI register group. The DMAC uses this base address as its destination address and copies the received frames to different areas.

Figure 9-5 Structure of the linked list

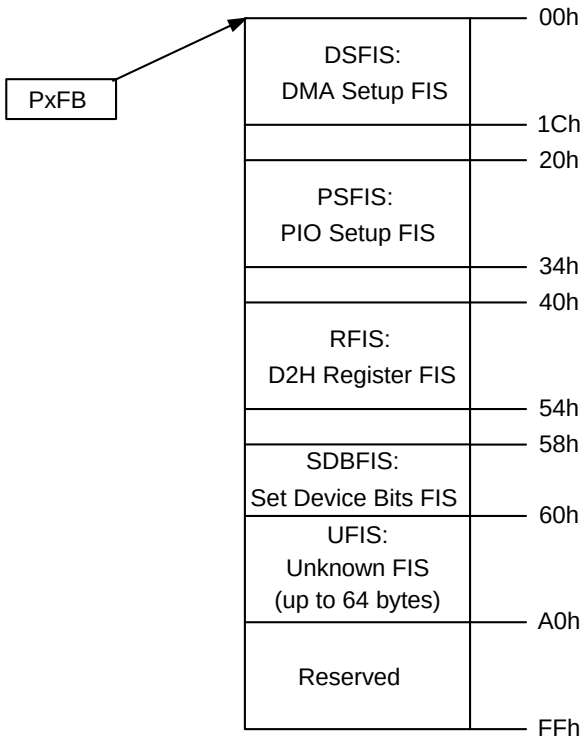
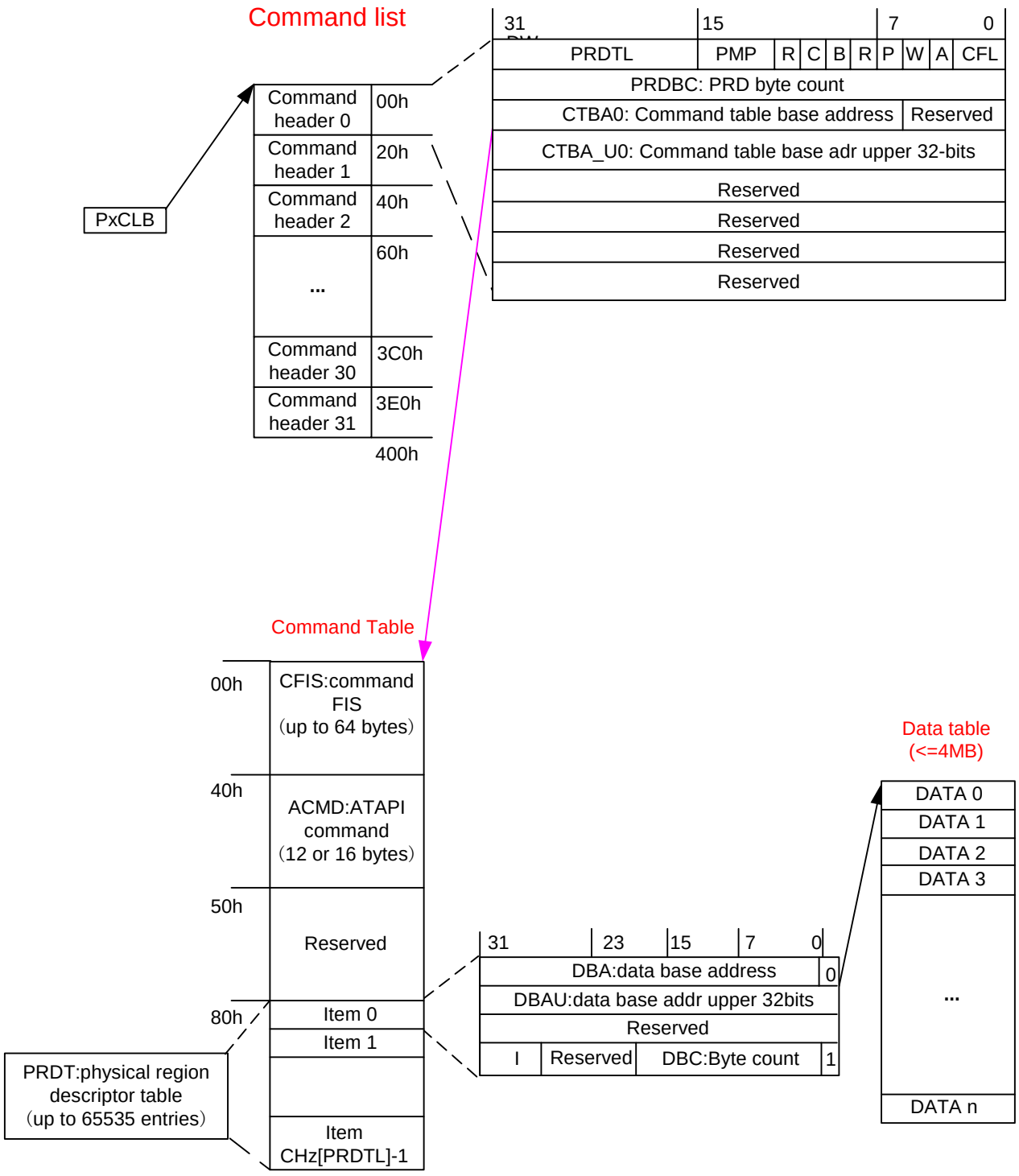


Figure 9-6 shows the structures of a command list and a data list. Such lists refer to the areas that are created in the system memory by the software. A command list contains up to 32 commands and its base address is specified by the PxCLB and PxCLBU registers in the AHCI register group. Each command has a command header in which the CTBA0 part specifies the base address of the command table. The command table contains the commands to be read and the data space linked list to be read and written.

Figure 9-6 Structures of the command list and data list



Each time before a command is executed, the preceding two lists must be created in the memory. For details about the meaning of the lists, see the AHCI1.2 protocol. In Figure 9-6 , the CFIS area is used to store the H2D register FISs. For details, see the SATA2.5 protocol. The ACMD area is used to store the commands relevant to the ATAPI operation. For details, see the protocols about the DVD devices and CD-ROMs provided by the Small Form Factor (SSF) Committee.

Contents

Figures

Tables

10 USB 2.0 Host

10.1 Overview

The USB 2.0 host controller supports the high-speed (480 Mbit/s), full-speed (12 Mbit/s), and low-speed (1.5 Mbit/s) data transfer modes. It also fully complies with the USB 2.0, open host controller interface (OHCI) Rev 1.0a, and enhanced host controller interface (EHCI) Rev 1.0 protocols. The USB 2.0 host controller consists of a root hub that is a part of the USB system and is used to extend the USB interface. Most of the hardware logics in the controller is used to:

- Control and process the data transfer.
- Parse data packets and packetize the data.
- Encode and decode the signals transmitted through the USB interface.
- Provide interfaces (such as the interrupt vector interface) for the driver.

10.2 Function Description

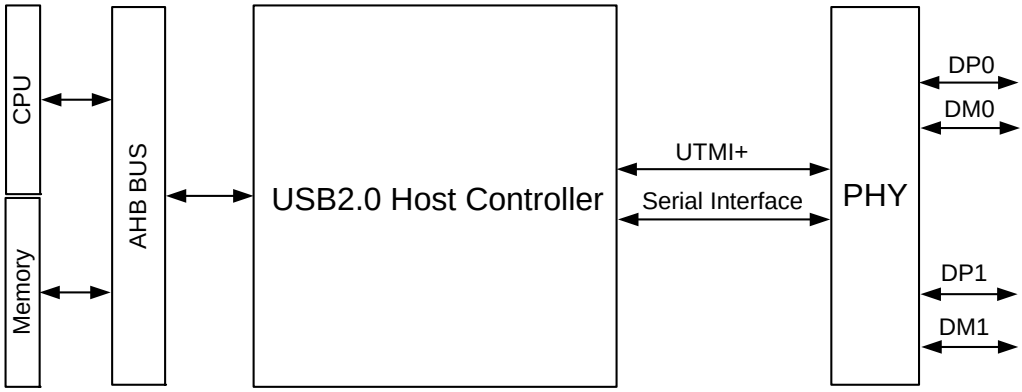
Features

The USB 2.0 host controller has the following features:

- Supports the USB 2.0 protocol.
- Supports the OHCI Rev 1.0a and EHCI Rev 1.0 protocols.
- Supports the high-speed, full-speed, and low-speed data transfer modes.
- Supports the low-power consumption solution.
- Supports four basic data transfer modes including control transfer, interrupt transfer, bulk transfer, and isochronous transfer.
- Supports the connections to up to 127 devices through USB hubs.

Figure 10-1 shows the logic block diagram of the USB module.

Figure 10-1 Logic block diagram of the USB module



UTMI: USB2.0 transceiver macrocell interface

Function Principle

The USB 2.0 host supports the following four standard transfer modes:

- Control transfer

  This mode applies to the data transfer between the USB host and the USB device endpoint 0. For the USB device with specific models, other endpoints can be used in control transfer mode. The control transfer is bi-directional, and the transferred data volume is small. Depending on the device and transfer speed, the 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. The device in this mode is the best choice when the transfer speed is low and many data transfers are delayed. In this case, bulk transfer is performed after all other types of data transfers are complete. This mode features in the error-free data transfer between the USB host and the device through error detection and re-transmission mechanism.

- Isochronous transfer

  This mode applies to the stream data transfer with the strict time requirement and strong error tolerance or the instant data transfer at a constant data rate. This mode provides a fixed bandwidth and time interval.

- Interrupt transfer

  This mode is designed to support those devices that do not need to send or receive data frequently but with bounded service periods. An interrupt pipe is a stream pipe and is therefore always uni-directional. An endpoint descriptor identifies whether the communication flow in a given interrupt pipe flows into or out of the host.

10.3 Operating Mode

10.3.1 Interface Signals

Table 10-1 lists the interface signals of the USB 2.0 host integrated with the USB physical layer entity sublayer (PHY).

Table 10-1 Interface signals of the USB 2.0 host

Signal Name	Direction	Function Description	Corresponding Pin
USBDP0	I/O	D+ data line of USB port 0	USBDP0
USBDM0	I/O	D- data line of USB port 0	USBDM0
USBDP1	I/O	D+ data line of USB port 1	USBDP1
USBDM1	I/O	D- data line of USB port 1	USBDM1
USBREXT	I/O	External resistor. The external reference impedance is 3.4 KΩ ± 1% and the external resistor is placed as close as possible to the USBREXT pin.	USBREXT
USBVDDA33T0_0	I/O	Analog power supply, 3.3 V ± 5%.	USBVDDA33T0_0
USBVDDA33T0_1	I/O	Analog power supply, 3.3 V ± 5%.	USBVDDA33T0_1
USBVSSA33T0_0	I/O	Analog ground.	USBVSSA33T0_0
USBVSSA33T0_1	I/O	Analog ground.	USBVSSA33T0_1
USBVSSA33T0_2	I/O	Analog ground.	USBVSSA33T0_2
USBVDDA33C	I/O	Analog power supply, 3.3 V ± 5%.	USBVDDA33C
USBVSSA33C	I/O	Analog ground.	USBVSSA33C
USBVDDA33T1_0	I/O	Analog power supply, 3.3 V ± 5%.	USBVDDA33T1_0
USBVDDA33T1_1	I/O	Analog power supply, 3.3 V ± 5%.	USBVDDA33T1_1
USBVSSA33T1_0	I/O	Analog ground.	USBVSSA33T1_0
USBVSSA33T1_1	I/O	Analog ground.	USBVSSA33T1_1
USBVSSA33T1_2	I/O	Analog ground.	USBVSSA33T1_2
USBVDD0	I/O	Digital power supply, 1.0 V ± 5%.	USBVDD0
USBVSS0	I/O	Digital ground.	USBVSS0
USBVDD1	I/O	Digital power supply, 1.0 V ± 5%.	USBVDD1
USBVSS1	I/O	Digital ground.	USBVSS1

10.3.2 Typical Application

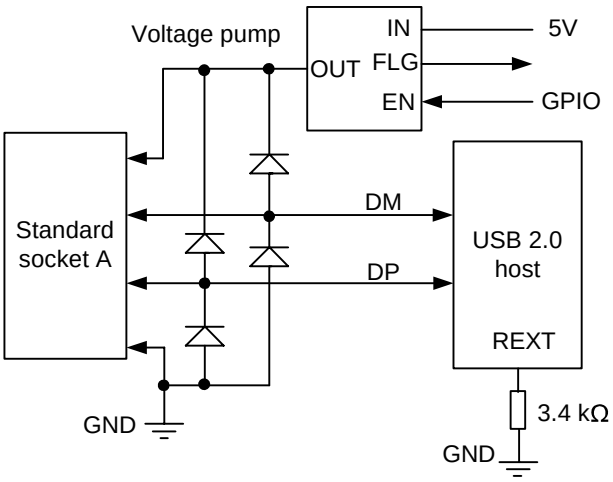
Figure 10-2 shows the reference design of the USB 2.0 host integrated with the PHY.

CAUTION

To ensure high signal quality, pay attention to the following points when designing the printed circuit board (PCB):

- A 45 Ω match resistor is configured for each of the DP and DM data line at the USB 2.0 host side. Therefore, no resistor needs to be connected externally.
- The capacitance of the external electrostatic discharge (ESD) device is about 1 pF.
- The EN signal and overcurrent protection of the voltage pump are controlled through the GPIO.

Figure 10-2 Reference design of the USB 2.0 host



10.3.3 Clock Gating

To enable the working clock of the controller, you need to write 1 to SC_PEREN[usb_clken]. To disable the working clock, you need to write 1 to SC_PERDIS[usb_clkdis]. After clock reset, the working clock is enabled by default.

10.3.4 Soft Reset

By default, the USB 2.0 host is always in reset state. To start the USB 2.0 host, you need to set the system controller SC_PERCTRL8[usb_srst] to 1 to clear the reset at the USB PHY side, and then set SC_PERCTRL8[usb_hrst] to 1 to clear the reset at the USB bus side. You also need to adjust the values of SC_PERCTRL12[usb_tune0] and SC_PERCTRL12[usb_tune1]. The recommended values are the standard level.

10.4 Register Summary

NOTE

The USB module is a standard USB 2.0 host and the internal registers are the standard EHCI and OHCI registers which are described in the EHCI and OHCI protocols. For details, see the related protocols. The following descriptions are related to the registers specially defined by the vendors.

Table 10-2 lists the USB registers.

Table 10-2 Summary of the USB registers (base address: 0x100B_0000)

Offset Address	Register	Description	Page
0x90	INTNREG00	Micro-frame length configuration register	10-5
0x94	INTNREG01	FIFO OUT/IN threshold register	10-6
0x98	INTNREG02	FIFO depth configuration register	10-6
0x9C	INTNREG03	Interrupt memory transfer enable register	10-7
0xA0	INTNREG04	Debug register	10-7
0xA4	INTNREG05	Control and status register	10-8
0xA8	INTNREG06	AHB error status register	10-10
0xAC	INTNREG07	AHB error address register	10-10

Note: The base address of the EHCI register is 0x100B_0000. The base address of the OHCI register is 0x100A_0000. The base address of the register listed in Table 10-2 is the same as that of the EHCI register.

10.5 Register Description

INTNREG00

INTNREG00 is the micro-frame length configuration register.

Offset Address	Register Name	Total Reset Value
0x90	INTNREG00	0x0000_0000

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

Bits	Access	Name	Description
[31:14]	-	reserved	Reserved.

[13:1]	RW	val	Micro-frame counter value. This register is used for emulation only. In normal state, the micro-frame length is 125 μs as the protocol specifies. In emulation, you can change the micro-frame length by configuring this register as required to reduce the emulation time.
[0]	RW	en	Register enable. 0: Disabled. 1: Enabled.

INTNREG01

INTNREG01 is the FIFO OUT/IN threshold register.

Offset Address	Register Name	Total Reset Value
0x94	INTNREG01	0x0020_0020

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

Bits	Access	Name	Description
[31:16]	RW	out_threshold	Transmit threshold. During data transmit, when the data amount in the FIFO is above the transmit threshold, data is transmitted.
[15:0]	RW	in_threshold	Receive threshold. During data receive, when the data amount in the FIFO is above the receive threshold, data is read from the FIFO.

INTNREG02

INTNREG02 is the FIFO depth configuration register.

Offset Address	Register Name	Total Reset Value
0x98	INTNREG02	0x0000_0080

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

Bits	Access	Name	Description
[31:12]	-	reserved	Reserved.
[11:0]	RW	fifo_depth	FIFO depth. The value defined here is 32 bits.

INTNREG03

INTNREG03 is the interrupt memory transfer enable register.

Offset Address	Register Name	Total Reset Value
0x9C	INTNREG03	0x0000_0001

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

Bits	Access	Name	Description
[31:1]	-	reserved	Reserved.
[0]	RO	brk_en	Interrupt memory transfer enable. 0: Disabled. 1: Enabled.

INTNREG04

INTNREG04 is the debug register.

Offset Address	Register Name	Total Reset Value
0xA0	INTNREG04	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																										auto_en	nak_reldfix_en	reserved	scaledwn_enum_time		hccparam_en	hcsparam_en
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:6]	-	reserved	Reserved.

[5]	RW	auto_en	Automatic feature enable. 0: The automatic feature is enabled. The suspend signal is deasserted when the run/stop bit is reset by software, but the hchalted bit is not set. 1: The automatic feature is disabled. The port is not suspended when software clears the run/stop bit. The default value is 0.
[4]	RW	nak_reldfix_en	NAK reload enable. 0: Enabled. 1: Disabled.
[3]	-	reserved	Reserved.
[2]	RW	scaledwn_enum_time	Port enumeration time scale down enable. 0: Disabled. 1: Enabled.
[1]	RW	hccparam_en	Write enable of the HCCPARAMS write register. 0: Disabled. 1: Enabled.
[0]	RW	hcsparam_en	Write enable of the HCSPARAMS register. 0: Disabled. 1: Enabled.

INTNREG05

INTNREG05 is the control and status register. It is used to read/write the PHY register.

The USB host interface can be configured as the UTMI or UTMI+ low pin interface (ULPI). The register descriptions vary according to interface types.

When the interface is a UTMI, the description is as follows.

Offset Address	Register Name	Total Reset Value
0xA4	INTNREG05	0x0000_1000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved														vbusy		vport			vcontrol_loadm	vcontrol				vstatus							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:18]	-	reserved	Reserved.

[17]	RO	vbusy	The value 1 indicates that hardware is writing data. This bit is cleared only when the process is complete.
[16:13]	RW	vport	Port ID which cannot exceed the supported number of ports.
[12]	RW	vcontrol_loadm	Load enable. 0: Enabled. 1: Disabled.
[11:8]	RW	vcontrol	Port control signal.
[7:0]	RO	vstatus	Port status signal.

When the interface is a ULPI, the description is as follows.

Offset Address	Register Name	Total Reset Value
0xA4	INTNREG05	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	vendor control	reserved			vport				access				immidiate address				extend address								value							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	R/W	vendor control	The register can be read/written by writing 1 to this bit. After the operation is complete, this bit is cleared automatically.
[30:28]	-	reserved	Reserved.
[27:24]	RW	vport	Port ID which cannot exceed the supported number of ports.
[23:22]	RW	access	Read/write access to the register. 00: Reserved. 01: Reserved. 10: Write access to the register. 11: Read access to the register.
[21:16]	RW	immidiate address	Register address.
[15:8]	RW	extend address	Extended register address.
[7:0]	RW	value	Register value. This value is written to the register during the write operation. This value is read from the register during the read operation after bit[31] is cleared.

INTNREG06

INTNREG06 is the AHB error status register.

Offset Address	Register Name	Total Reset Value
0xA8	INTNREG06	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	err_capture	reserved																			hbusrt_err			num_beat_err					num_beat_ok			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31]	RW	err_capture	Indicates that an AHB error is captured.
[30:12]	-	reserved	Reserved.
[11:9]	RO	hbusrt_err	Indicates the hbrust value during a control transfer when an AHB error occurs.
[8:4]	RO	num_beat_err	Indicates the number of beats during a burst transfer when an AHB error occurs. The maximum number of beats is 16. 0x00–0x10: Valid. 0x11–0x1F: Reserved.
[3:0]	RO	num_beat_ok	Indicates the number of beats that are complete successfully during a burst transfer when an AHB error occurs.

INTNREG07

INTNREG07 is the AHB error address register.

Offset Address	Register Name	Total Reset Value
0xAC	INTNREG07	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	err_addr	Indicates the address during a control transfer when an AHB error occurs.

Contents

Figures

Tables

11 Other Peripheral Interfaces

11.1 I²C Interface

11.1.1 Overview

The inter-integrated circuit (I^2C) interface implements the standard I^2C master/slave functions and complies with the Philips I^2C bus protocol. It can serve as a master to receive/transmit data from/to the slave on the I^2C bus or serve as a slave to respond to the data transmit/receive request sent from the master. The I^2C interface is mainly used for controlling external I^2C devices, such as analog-to-digital converter (ADC) and digital-to-analog converter (DAC).

11.1.2 Features

The I^2C interface has the following features:

- Supports the standard I^2C bus protocol.
- Supports master and slave operations.
- Supports 7-bit and 10-bit device addresses.
- Supports the programmable clock that controls the communication rate.
- Supports the directory memory access (DMA) interface.
- Supports the interrupt and query operation modes.

11.1.3 Signal Description

Table 11-1 lists the signals of the I^2C interface.

CAUTION

The serial data (SDA) and serial clock (SCL) I^2C pins comply with the I^2C specification in OD (open drain) connection mode.

Table 11-1 Signals of the I²C interface

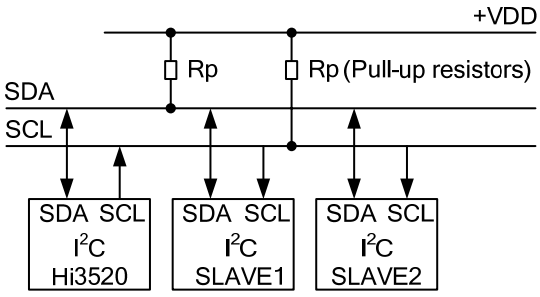
Signal Name	Direction	Description	Corresponding Pin
I2C_SDA	I/O	I²C bidirectional data signal, multiplexed with the GPIO. For details about the multiplexing configuration information, see section 11.1.5 "Pin Multiplexing."	SDA
I2C_SCL	I/O	I²C clock signal, multiplexed with the GPIO. For details about the multiplexing configuration information, see section 11.1.5 "Pin Multiplexing."	SDL

11.1.4 Function Description

Typical Application

Figure 11-1 shows the circuit diagram of the typical I²C bus application.

Figure 11-1 Circuit diagram of the typical I²C application



The I²C bus is a two-wire and bidirectional serial bus that implements a simple and effective data transfer. This bus simplifies the connections between devices and is applicable to the transfer of a small amount of data between multiple short-distance devices. With the flexibility of the I²C bus, system development and device extension are implemented easily.

Function Principle

A typical I²C data transfer involves the start signal, slave address transmit, data transfer, and stop signal. Figure 11-2 shows the timing and Figure 11-3 shows the data frame format.

Figure 11-2 Timing for I²C data transfer

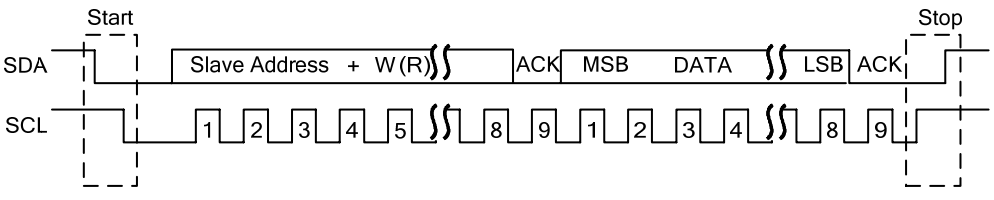
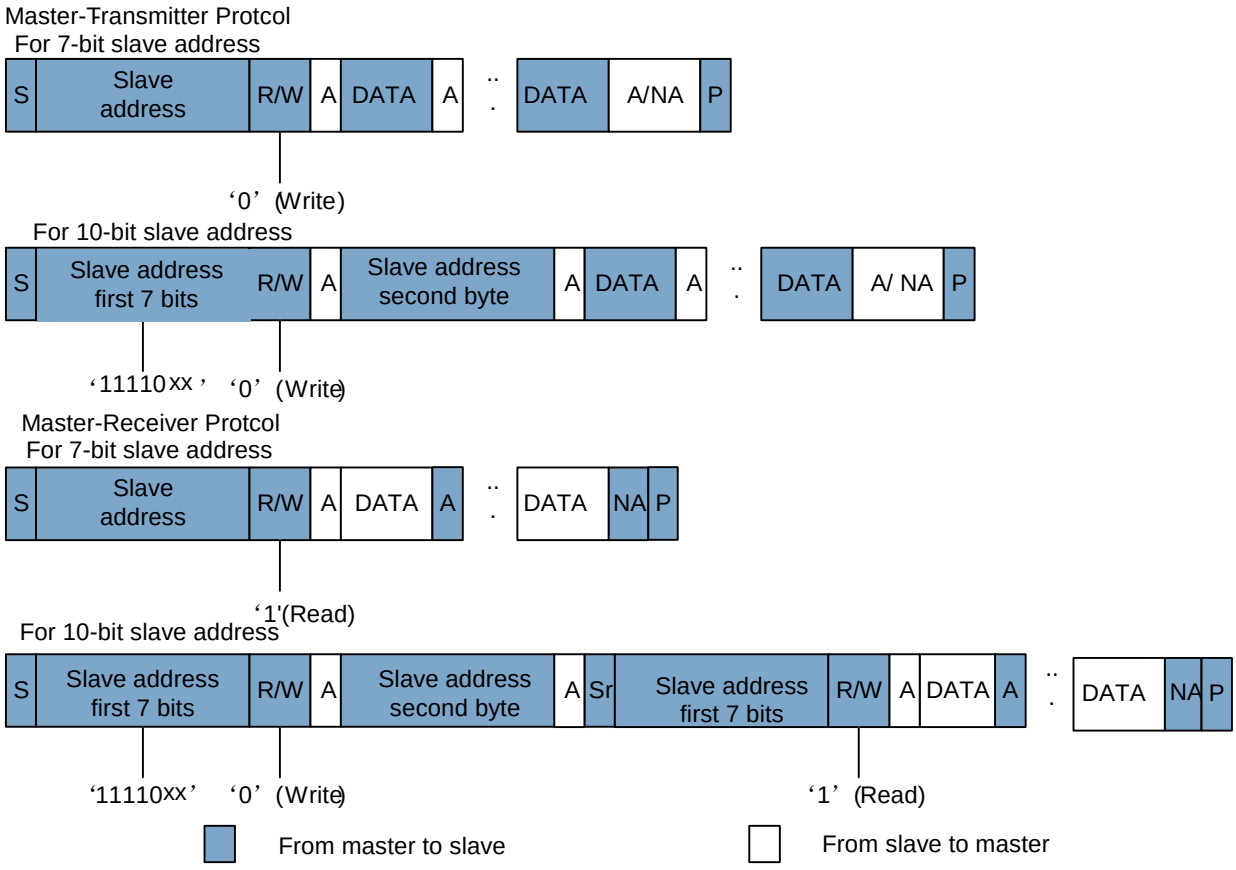


Figure 11-3 Frame format for I²C data transfer



A	Acknowledge (SDA low)
NA	No acknowledge (SDA high)
S	Start condition
Sr	Repeated start condition
P	Stop condition

According to the standard I²C protocol, the start signal (Start) is a special signal sent by the master on the bus to wake up all the slaves and indicate the start of the data transfer.

The slave address transmit (slave address + W/R) refers to the first data packet sent by the master after the start condition. It consists of seven slave address bits and one read/write bit. If it is a 10-bit slave address, two data packets need to be transmitted. In addition, other bits (except the 10 bits of address) are reserved according to the protocol. Based on the address information, each slave determines whether to transmit or receive data. After the slave address is successfully received, the corresponding slave pulls down the SDA at the ninth SCL cycle and returns an acknowledge signal (ACK) to the master. Then, the subsequent data transfer is started.

The data transfer (DATA) refers to data receive or transmit according to read and write commands after the master successfully receives the slave address ACK signal. During the data transfer, the SDA changes its state only when the SCL is low. The SDA holds when the SCL is high. Each time the receiver receives one byte, it must send an ACK signal to the

transmitter. If the transmitter does not successfully receive the ACK signal, it stops the data transfer or starts the transfer again.

The stop signal (Stop) is a special signal sent by the master to indicate the end of the data transfer according to the I^2C protocol when the current data transfer is complete and no subsequent data transfer needs to be initiated yet. When the master sends the stop signal, the slave must release the bus.

11.1.5 Operating Mode

Pin Multiplexing

The SDA I^2C pin is multiplexed with GPIO0_0 and the SCL I^2C pin is multiplexed with GPIO0_1. You can select the pin by configuring the IO Config registers reg37 and reg38.

Clock Configuration

In I^2C master mode, the high and low level widths of the SCL I^2C bus signal at standard mode and high-speed mode, namely, the number of cycles relative to the I^2C working clock, can be set through the registers such as I2C_SS_SCL_HCNT, I2C_SS_SCL_LCNT, I2C_FS_SCL_HCNT, and I2C_FS_SCL_LCNT can be respectively calculated by the formulas:

$$*\_HCNT = T_{scl\_h} \times F_{I2C} - 8 \quad \text{and} \quad *\_LCNT = T_{scl\_l} \times F_{I2C} - 1$$

NOTE

- T_{scl_h} indicates the high level width of the SCL signal (in cycle).
- T_{scl_l} indicates the low level width of the SCL signal (in μs).
- F_{I2C} indicates the frequency of the I^2C working clock (in MHz).
- In I^2C slave mode, the configurations of the registers I2C_SS_SCL_HCNT, I2C_SS_SCL_LCNT, I2C_FS_SCL_HCNT and I2C_FS_SCL_LCNT are not required.

Table 11-2 to Table 11-5 list the typical values of I2C_SS_SCL_HCNT, I2C_SS_SCL_LCNT, I2C_FS_SCL_HCNT and I2C_FS_SCL_LCNT.

Table 11-2 Typical values of I2C_SS_SCL_HCNT

I²C Bus Rate (kbit/s)	I²C Working Clock (MHz)	SCL High Level Width (μs)	I2C_SS_SCL_HCNT (Cycle)
100	100	4	400

Table 11-3 Typical values of I2C_SS_SCL_LCNT

I²C Bus Rate (kbit/s)	I²C Working Clock (MHz)	SCL Low Level Width (μs)	I2C_SS_SCL_LCNT (Cycle)
100	100	4.7	470

Table 11-4 Typical values of I2C_FS_SCL_HCNT

I²C Bus Rate (kbit/s)	I²C Working Clock (MHz)	SCL High Level Width (μs)	I2C_FS_SCL_HCNT (Cycle)
400	100	0.6	60

Table 11-5 Typical values of I2C_FS_SCL_LCNT

I²C Bus Rate (kbit/s)	I²C Working Clock (MHz)	SCL Low Level Width (μs)	I2C_FS_SCL_LCNT (Cycle)
400	100	1.3	130

Soft Reset

The I^{2}C controller can be reset separately by configuring SC_PERCTRL8[i2c_srst]. After the reset, the configuration registers are set to the default values. Therefore, these registers must be initialized again.

Data Transfer in Interrupt or Query Mode

The initialization process in master mode is as follows:

Step 1 Set the timeout flag and detect I2C_STATUS[activity] to check whether the I^{2}C controller is idle. If I2C_STATUS[activity] is 1, wait and continue the detection until I2C_STATUS[activity] becomes 0 or the timeout occurs. Then set I2C_ENABLE[enable] to 0 to disable the I^{2}C controller.

Step 2 Write corresponding values to I2C_CON to configure the parameters such as the master mode and transfer rate.

Step 3 Write the slave address of the interconnected device to I2C_TAR[i2c_tar].

Step 4 Configure I2C_SS_SCL_HCNT and I2C_SS_SCL_LCNT to set the I^{2}C bus clock cycles.

Step 5 Configure I2C_RX_TL and I2C_TX_TL to set the thresholds of TX_FIFO and RX_FIFO.

Step 6 If the driver runs in interrupt mode, set I2C_INTR_MASK to enable the corresponding interrupts. If in query mode, disable the corresponding interrupts.

Step 7 Write 1 to I2C_ENABLE[enable] to enable the I^{2}C. The initialization configuration is complete.

----End

The initialization process in slave mode is as follows:

Step 1 Set the timeout flag and detect I2C_STATUS[activity] to check whether the I^{2}C controller is idle. If I2C_STATUS[activity] is 1, wait and continue the detection until I2C_STATUS[activity] becomes 0 or the timeout occurs. Then, write 0 to I2C_STATUS[enable] to disable the I^{2}C controller.

Step 2 Write corresponding values to I2C_CON to configure the parameters such as the slave mode and transfer rate.

Step 3 Write the responded slave address in slave mode to I2C_SAR[i2c_sar].

Step 4 Configure I2C_RX_TL and I2C_TX_TL to set the thresholds of TX_FIFO and RX_FIFO.

Step 5 If the driver runs in interrupt mode, set I2C_INTR_MASK to enable the corresponding interrupts. If in query mode, disable the corresponding interrupts.

Step 6 Write 1 to I2C_ENABLE[enable] to enable the I^2C. The initialization configuration is complete.

----End

To transmit data in master mode, do as follows:

Step 1 Write the data to be transmitted to I2C_DATA_CMD and start data transmit.

Step 2 In query mode, check the TX_FIFO status by reading I2C_STATUS and I2C_TXFLR during continuous data transmit. In interrupt mode, check the corresponding interrupt status bits. Before data transmit is complete, ensure that the data in the TX_FIFO does not overflow (otherwise, data is lost) or is not empty (otherwise, the I^2C considers that the transfer ends and sends the stop signal).

Step 3 Check whether I2C_STATUS[tfe] is 1. If it is 1, it indicates that data transmit is complete. If it is 0, it indicates that data transmit is not complete and detection continues until 1 is detected.

----End

To transmit data in slave mode, do as follows:

Step 1 When other masters on the I^2C bus start to receive data and the address matches the value of I2C_SAR, the I^2C enables I2C_RAW_INTR_STAT[r_rd_req]. Then, the interrupt is enabled in interrupt mode.

Step 2 In query mode, if software detects that I2C_RAW_INTR_STAT[r_rd_req] is valid, it writes the data to be transmitted to I2C_DATA_CMD. In interrupt mode, if the raw interrupt RD_REQ is found through the detection of corresponding interrupt status bit, then the data to be transmitted is written to I2C_DATA_CMD.

Step 3 Check whether I2C_STATUS[tfe] is 1. If it is 1, it indicates that data transmit is complete. If it is 0, it indicates that data transmit is not complete and detection continues until 1 is detected.

----End

To receive data in master mode, do as follows:

Step 1 Write the read command (0x0100) to I2C_DATA_CMD to start data receive.

Step 2 When receiving data continuously, send the read command (0x0100) to I2C_DATA_CMD for several times. The send times must be equal to the number of data segments to be received. For example, to receive three data segments, send the read command to the I2C_DATA_CMD register for three times. Before data receive is complete, ensure that the data (namely, the read command 0x100) in TX_FIFO does not overflow or the FIFO is not empty. In addition, the RX_FIFO status must be detected during continuous data receive to

avoid RX_FIFO overflow. In interrupt mode, the RX_FIFO status can be queried by detecting the interrupt status bit.

Step 3 Check whether I2C_STATUS[activity] is 0. If it is 0, it indicates that data receive is complete. If it is 1, it indicates that data receive is not complete and detection continues until 0 is detected.

----End

To receive data in slave mode, do as follows:

Step 4 Other masters on the I^2C bus start to receive data and the operation address matches with the value of I2C_SAR.

Step 5 The I^2C interface receives data and stores it in RX_FIFO. During continuous data receive, the RX_FIFO status must be detected to avoid RX_FIFO overflow. In interrupt mode, the RX_FIFO status can be queried by detecting the interrupt status bit.

Step 6 Check whether I2C_RAW_INTR_STAT[r_rx_done] is 1. If it is 1, it indicates that data receive is complete. If it is 0, it indicates that data receive is not complete and detection continues until 1 is detected.

----End

Data Transfer in DMA Mode

The initialization process is as follows:

Step 1 Set the timeout flag and detect I2C_STATUS[activity] to check whether the I^2C controller is idle. If I2C_STATUS[activity] is 1, wait and continue the detection until I2C_STATUS[activity] becomes 0 or the timeout occurs. Then, write 0 to I2C_STATUS[enable] to disable the I^2C controller.

Step 2 Write corresponding values to I2C_CON to configure the parameters such as the master mode and transfer rate.

Step 3 Write the slave address of the interconnected device to I2C_TAR[i2c_tar] in master mode.

Step 4 Configure I2C_SS_SCL_HCNT and I2C_SS_SCL_LCNT to set the I^2C bus clock cycles.

Step 5 Configure I2C_DMA_TDLR and I2C_DMA_RDLR to set the thresholds of the DMA TX_FIFO and RX_FIFO.

Step 6 Write 1 to I2C_ENABLE[enable] to enable the I^2C. The initialization configuration is complete.

----End

To transmit data, do as follows:

Step 1 Configure the DMA data channel, including the source and destination addresses, number of data segments to be transmitted, and transfer type.

Step 2 Set I2C_DMA_CR to 0x2 to enable the DMA transmit function of the I^2C.

Step 3 Check whether the data is transmitted completely by querying the DMA interrupt status. If data transmit is complete, disable the DMA transmit function of the I^2C.

----End

Data receive consists involves two steps: data transmit and data receive. That is, send the read command to the I²C controller and then read data from RX_FIFO. To receive data, do as follows:

Step 1 Configure the DMA data channel, including the parameters of data transfer source (containing the read command 0x100) and destination address, address of data receive area, number of data segments to be transferred, and transfer type.

Step 2 Set I2C_DMA_CR to 0x0003 to enable DMA transmit and receive functions of the I²C. The DMA transmit function needs to be enabled because the read command 0x100 must be sent to I2C_DATA_CMD before data is received.

Step 3 Check whether the data is received completely by querying the DMA interrupt status. If data receive is complete, disable the DMA receive function of the I²C.

----End

11.1.6 Register Summary

Table 11-6 Summary of the I²C registers (base address: 0x200D_0000)

Offset Address	Register	Description	Page
0x0000	I2C_CON	I²C control register	11-9
0x0004	I2C_TAR	I²C access slave address register	11-10
0x0008	I2C_SAR	Slave I²C address register	11-11
0x0010	I2C_DATA_CMD	I²C data channel register	11-12
0x0014	I2C_SS_SCL_HCNT	High level width configuration register for the SCL clock at standard speed	11-12
0x0018	I2C_SS_SCL_LCNT	Low level width configuration register for the SCL clock at standard speed	11-13
0x001C	I2C_FS_SCL_HCNT	High level width configuration register for the SCL clock at a high speed	11-14
0x0020	I2C_FS_SCL_LCNT	Low level width configuration register for the SCL clock at a high speed	11-14
0x002C	I2C_INTR_STAT	Interrupt status register	11-15
0x0030	I2C_INTR_MASK	Interrupt mask register	11-17
0x0034	I2C_RAW_INTR_STAT	Raw interrupt status register	11-18
0x0038	I2C_RX_TL	RX_FIFO threshold configuration register	11-19
0x003C	I2C_TX_TL	TX_FIFO threshold configuration register	11-20
0x0040	I2C_CLR_INTR	Combined and independent interrupt clear register	11-20

Offset Address	Register	Description	Page
0x0044	I2C_CLR_RX_UNDER	RX_UNDER interrupt clear register	11-21
0x0048	I2C_CLR_RX_OVER	RX_OVER interrupt clear register	11-21
0x004C	I2C_CLR_TX_OVER	TX_OVER interrupt clear register	11-22
0x0054	I2C_CLR_TX_ABRT	ABRT interrupt clear register	11-22
0x005C	I2C_CLR_ACTIVITY	ACTIVITY status register	11-23
0x0060	I2C_CLR_STOP_DET	STOP_DET interrupt clear register	11-23
0x0064	I2C_CLR_START_DET	START_DET interrupt clear register	11-24
0x0068	I2C_CLR_GEN_CALL	GEN_CALL interrupt clear register	11-24
0x006C	I2C_ENABLE	I^2C enable register	11-25
0x0070	I2C_STATUS	I^2C status register	11-25
0x0074	I2C_TXFLR	TX_FIFO data count indication register	11-26
0x0078	I2C_RXFLR	RX_FIFO data count indication register	11-27
0x0080	I2C_TX_ABRT_SOURCE	TX_ABRT source interrupt register	11-27
0x0088	I2C_DMA_CR	I^2C DMA channel enable control register	11-29
0x008C	I2C_DMA_TDLR	TX_FIFO threshold configuration register for DMA operation	11-30
0x0090	I2C_DMA_RDLR	RX_FIFO threshold configuration register for DMA operation	11-30

11.1.7 Register Description

I2C_CON

I2C_CON is the I^2C control register.

CAUTION

I2C_CON can be configured only when the I^2C is disabled, that is, I2C_ENABLE[enable] is 0.

Offset Address	Register Name	Total Reset Value
0x0000	I2C_CON	0x0075

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved									reserved	i2c_restart_en	i2c_10bitaddr_master	reserved	Speed		master_mode
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1

Bits	Access	Name	Description
[15:7]	RO	reserved	Reserved.
[6]	RW	reserved	Reserved. The bit must be 1.
[5]	RW	i2c_restart_en	Restart condition enable. 0: The restart condition is disabled. 1: The restart condition is enabled.
[4]	RW	i2c_10bitaddr_master	7-bit or 10-bit address select. 0: The 7-bit address is selected. 1: The 10-bit address is selected.
[3]	RO	reserved	Reserved.
[2:1]	RW	speed	I^2C operation speed select. 00: Invalid speed. 01: Standard speed, namely, 100 kbit/s. 10: High speed, namely, 400kbit/s. 11: Reserved. Note that if the bit is set to 00 or 11, it assumes that the bit is set to 10.
[0]	RW	master_mode	Master mode enable. 0: The master mode is disabled. 1: The Master mode is enabled.

I2C_TAR

I2C_TAR is the I^2C access slave address register.

CAUTION

I2C_TAR can be configured only when the I²C is disabled, that is, I2C_ENABLE[enable] is 0.

Offset Address	Register Name	Total Reset Value
0x0004	I2C_TAR	0x009C

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				special	gc_or_start	i2c_tar									
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0

Bits	Access	Name	Description
[15:12]	RO	reserved	Reserved.
[11]	RW	special	General call and start byte functions enable. 0: The functions are disabled. 1: The function are enabled.
[10]	RW	gc_or_start	This bit determines whether the function to be implemented is general call or start byte when bit[11] is 1. 0: The function to be implemented is general call. 1: The function to be implemented is start byte.
[9:0]	RW	i2c_tar	Slave address that is accessed by the I²C when I²C acts as a master.

I2C_SAR

I2C_SAR is the Slave I²C address register.

CAUTION

I2C_SAR can be configured only when the I²C is disabled, that is, I2C_ENABLE[enable] is 0.

Offset Address	Register Name	Total Reset Value
0x0008	I2C_SAR	0x0055

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								i2c_sar							
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

Bits	Access	Name	Description
[15:10]	RO	reserved	Reserved.
[9:0]	RW	i2c_sar	Address of the slave I^2C.

I2C_DATA_CMD

I2C_DATA_CMD is the I^2C data channel register.

Offset Address	Register Name	Total Reset Value
0x0010	I2C_DATA_CMD	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved							cmd	dat							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:9]	RO	reserved	Reserved.
[8]	RW	cmd	Read/write control bit. 0: Write operation. It indicates that the I^2C controller will transmit data to the I^2C bus. In this case, the eight LSBs (DAT) are to be transmitted to the I^2C bus by the I^2C controller. 1: Read operation. It indicates that the I^2C controller will read data from the I^2C bus.
[7:0]	RW	dat	Data to be transmitted or received through the I^2C bus. In read operation, the eight bits are the data received on the I^2C bus; in write operation, the eight bits are the data transmitted on the I^2C bus.

I2C_SS_SCL_HCNT

I2C_SS_SCL_HCNT is the high level width configuration register for the SCL clock at standard speed.

CAUTION

I2C_SS_SCL_HCNT can be configured only when the I²C is disabled, that is, I2C_ENABLE[enable] is 0.

Offset Address	Register Name	Total Reset Value
0x0014	I2C_SS_SCL_HCNT	0x007A

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i2c_ss_scl_hcnt															
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0

Bits	Access	Name	Description
[15:0]	RW	i2c_ss_scl_hcnt	For details about how to calculate the high level width of the SCL clock at standard speed, see 11.1.5 "Clock Configuration." Note that the minimum value is 6 and any written value less than 6 is regarded as 6.

I2C_SS_SCL_LCNT

I2C_SS_SCL_LCNT is the low level width configuration register for the SCL clock at standard speed.

CAUTION

I2C_SS_SCL_HCNT can be configured only when the I²C is disabled, that is, I2C_ENABLE[enable] is 0.

Offset Address	Register Name	Total Reset Value
0x0018	I2C_SS_SCL_LCNT	0x008F

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i2c_ss_scl_lcnt															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1

Bits	Access	Name	Description
[15:0]	RW	i2c_ss_scl_lcnt	For details about how to calculate the low level width of the SCL clock at standard speed, see 11.1.5 "Clock Configuration." Note that the minimum value is 8 and any written value less than 8 is regarded as 8.

I2C_FS_SCL_HCNT

I2C_FS_SCL_HCNT is the high level width configuration register for the SCL clock at a high speed.

CAUTION

I2C_FS_SCL_HCNT can be configured only when the I^{2}C is disabled, that is, I2C_ENABLE[enable] is 0.

Offset Address	Register Name	Total Reset Value
0x001C	I2C_FS_SCL_HCNT	0x0013

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i2c_fs_scl_hcnt															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Bits	Access	Name	Description
[15:0]	RW	i2c_fs_scl_hcnt	For details about how to calculate the high level width of the SCL clock at a high speed, see 11.1.5 "Clock Configuration." Note that the minimum value is 6 and any written value less than 6 is regarded as 6.

I2C_FS_SCL_LCNT

I2C_FS_SCL_LCNT is the low level width configuration register for the SCL clock at a high speed.

CAUTION

I2C_FS_SCL_LCNT can be configured only when the I^{2}C is disabled, that is, I2C_ENABLE[enable] is 0.

Offset Address	Register Name	Total Reset Value
0x0020	I2C_FS_SCL_LCNT	0x0028

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	i2c_fs_scl_lcnt															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Bits	Access	Name	Description
[15:0]	RW	i2c_fs_scl_lcnt	For details about how to calculate the low level width of the SCL clock at a high speed, see 11.1.5 "Clock Configuration." Note that the minimum value is 8 and any written value less than 8 is regarded as 8.

I2C_INTR_STAT

I2C_INTR_STAT is the interrupt status register.

Offset Address	Register Name	Total Reset Value
0x002C	I2C_INTR_STAT	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				gen_call	tart_det	stop_det	activity	rx_done	tx_abrt	rd_req	tx_empty	tx_over	rx_full	rx_over	rx_under
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:12]	RO	reserved	Reserved.
[11]	RO	gen_call	GEN_CALL interrupt. It indicates whether a general call request is received. 0: No general call request is received. 1: A general call request is received. The I^2C stores the received data in the RX buffer.
[10]	RO	start_det	START_DET interrupt. It indicates whether the start condition is met on the I^2C bus. 0: The start condition is not met on the I^2C bus. 1: The start condition is met on the I^2C bus.
[9]	RO	stop_det	STOP_DET interrupt. It indicates whether the stop condition is met on the I^2C bus. 0: The stop condition is not met on the I^2C bus. 1: The stop condition is met on the I^2C bus.

[8]	RO	activity	ACTIVITY interrupt. It indicates the activity status of the I^2C. 0: The I^2C is idle. 1: The I^2C is busy.
[7]	RO	rx_done	RX_DONE interrupt. When the I^2C acts as the slave, this bit indicates whether data receive is complete. 0: Data receive is not complete. 1: Data receive is complete.
[6]	RO	tx_abrt	TX_ABRT interrupt. The interrupt can be triggered in multiple cases. For details, see I2C_TX_ABRT_SOURCE.
[5]	RO	rd_req	RD_REQ interrupt. When the I^2C acts as the slave, this bit indicates whether a data read request is initiated by a master. 0: No data read request is initiated by a master. 1: A data read request is initiated by a master.
[4]	RO	tx_empty	ITX_EMPTY interrupt. It indicates whether the data in the TX_FIFO reaches or is below the threshold. 0: The data in the TX_FIFO is above the threshold. 1: The data in the TX_FIFO reaches or is below the threshold.
[3]	RO	tx_over	TX_OVER interrupt. It indicates whether the data in TX_FIFO overflows. 0: The data in TX_FIFO does not overflow. 1: The data in TX_FIFO overflows.
[2]	RO	rx_full	RX_FULL interrupt. It indicates whether the data in RX_FIF reaches or is below the threshold. 0: The data in RX_FIF is below the threshold 1: The data in RX_FIF reaches or is above the threshold.
[1]	RO	rx_over	RX_OVER interrupt. It indicates whether the data in RX_FIFO overflows. 0: The data in RX_FIFO does not overflow. 1: The data in RX_FIFO overflows.

[0]	RO	rx_under	RX_UNDER interrupt. When RX_FIFO is empty, the internal bus interface initiates a request for reading I2C_DATA_CMD. 0: Meaningless. 1: When RX_FIFO is empty, the CPU reads I2C_DATA_CMD.

I2C_INTR_MASK

I2C_INTR_MASK is the interrupt mask register.

Offset Address	Register Name	Total Reset Value
0x0030	I2C_INTR_MASK	0x08FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				m_gen_call	m_start_det	m_stop_det	m_activity	m_rx_done	m_tx_abrt	m_rd_req	m_tx_empty	m_tx_over	m_rx_full	m_rx_over	m_rx_under
Reset	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1

Bits	Access	Name	Description
[15:12]	RO	reserved	Reserved.
[11]	RW	m_gen_call	GEN_CALL interrupt mask. 0: The GEN_CALL interrupt is masked. 1: The GEN_CALL interrupt is not masked.
[10]	RW	m_start_det	START_DET interrupt mask. 0: The START_DET interrupt is masked. 1: The START_DET interrupt is not masked.
[9]	RW	m_stop_det	STOP_DET interrupt mask. 0: The STOP_DET interrupt is masked. 1: The STOP_DET interrupt is not masked.
[8]	RW	m_activity	ACTIVITY interrupt mask. 0: The ACTIVITY interrupt is masked. 1: The ACTIVITY interrupt is not masked.
[7]	RW	m_rx_done	RX_DONE interrupt mask. 0: The RX_DONE interrupt is masked. 1: The RX_DONE interrupt is not masked.
[6]	RW	m_tx_abrt	TX_ABRT interrupt mask. 0: The TX_ABRT interrupt is masked. 1: The TX_ABRT interrupt is not masked.

[5]	RW	m_rd_req	RD_REQ interrupt mask. 0: The RD_REQ interrupt is masked. 1: The RD_REQ interrupt is not masked.
[4]	RW	m_tx_empty	TX_EMPTY interrupt mask. 0: The TX_EMPTY interrupt is masked. 1: The TX_EMPTY interrupt is not masked.
[3]	RW	m_tx_over	TX_OVER interrupt mask. 0: The TX_OVER interrupt is masked. 1: The TX_OVER interrupt is not masked.
[2]	RW	m_rx_full	RX_FULL interrupt mask. 0: The RX_FULL interrupt is masked. 1: The RX_FULL interrupt is not masked.
[1]	RW	m_rx_over	RX_OVER interrupt mask. 0: The RX_OVER interrupt is masked. 1: The RX_OVER interrupt is not masked.
[0]	RW	m_rx_under	RX_UNDER interrupt mask. 0: The RX_UNDER interrupt is masked. 1: The RX_UNDER interrupt is not masked.

I2C_RAW_INTR_STAT

I2C_RAW_INTR_STAT is the raw interrupt status register.

Offset Address	Register Name	Total Reset Value
0x0034	I2C_RAW_INTR_STAT	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				r_gen_call	r_start_det	r_stop_det	r_activity	r_rx_done	r_tx_abrt	r_rd_req	r_tx_empty	r_tx_over	r_rx_full	r_rx_over	r_rx_under
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:12]	RO	reserved	Reserved.
[11]	RO	r_gen_call	GEN_CALL raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.

[10]	RO	r_start_det	START_DET raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[9]	RO	r_stop_det	STOP_DET raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	r_activity	ACTIVITY raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	r_rx_done	RX_DONE raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[6]	RO	r_tx_abrt	Raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	r_rd_req	RD_REQ raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	r_tx_empty	R_TX_EMPTY raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[3]	RO	r_tx_over	TX_OVER raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	r_rx_full	RX_FULL raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	r_rx_over	RX_OVER raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	r_rx_under	RX_UNDER raw interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.

I2C_RX_TL

I2C_RX_TL is the RX_FIFO threshold configuration register.

Offset Address	Register Name	Total Reset Value
0x0038	I2C_RX_TL	0x0003

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								rx_tl							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	Access	Name	Description
[15:8]	RO	reserved	Reserved.
[7:0]	RW	rx_tl	RX_FIFO threshold. The actual value is equal to the configured value plus 1. Note that any configured value greater than 8 is considered as 8.

I2C_TX_TL

I2C_TX_TL is the TX_FIFO threshold configuration register.

Offset Address	Register Name	Total Reset Value
0x003C	I2C_TX_TL	0x0003

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved								tx_tl							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	Access	Name	Description
[15:8]	RW	Reserved	Reserved.
[7:0]	RW	tx_tl	TX_FIFO threshold. Note that any configured value greater than 8 is considered as 8.

I2C_CLR_INTR

I2C_CLR_INTR is the combined and independent interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x0040	I2C_CLR_INTR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_intr
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_intr	When this register is read, all combined interrupts and independent interrupts as well as the I2C_TX_ABRT_SOURCE register are cleared. Note that I2C_TX_ABRT_SOURCE[abrt_sbyte_norstrt] and the triggered combined interrupts cannot be cleared.

I2C_CLR_RX_UNDER

I2C_CLR_RX_UNDER is the RX_UNDER interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x0044	I2C_CLR_RX_UNDER	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_rx_under
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_rx_under	When this register is read, the RX_UNDER interrupt is cleared.

I2C_CLR_RX_OVER

I2C_CLR_RX_OVER is the RX_OVER interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x0048	I2C_CLR_RX_OVER	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_rx_over
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_rx_over	When this register is read, the RX_OVER interrupt is cleared.

I2C_CLR_TX_OVER

I2C_CLR_TX_OVER is the TX_OVER interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x004C	I2C_CLR_TX_OVER	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_tx_over
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_tx_over	When this register is read, the TX_OVER interrupt is cleared.

I2C_CLR_TX_ABRT

I2C_CLR_TX_ABRT is the ABRT interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x0054	I2C_CLR_TX_ABRT	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_tx_abrt
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_tx_abrt	When this register is read, the TX_ABRT interrupt and the I2C_TX_ABRT_SOURCE register are cleared.

I2C_CLR_ACTIVITY

I2C_CLR_ACTIVITY is the ACTIVITY status register.

Offset Address	Register Name	Total Reset Value
0x005C	I2C_CLR_ACTIVITY	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_activity
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_activity	When this register is read, the ACTIVITY interrupt status can be queried and hardware is cleared automatically.

I2C_CLR_STOP_DET

I2C_CLR_STOP_DET is the STOP_DET interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x0060	I2C_CLR_STOP_DET	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_stop_det
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_stop_det	When this register is read, the STOP_DET interrupt is cleared.

I2C_CLR_START_DET

I2C_CLR_START_DET is the START_DET interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x0064	I2C_CLR_START_DET	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_start_det
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_start_det	When this register is read, the START_DET interrupt is cleared.

I2C_CLR_GEN_CALL

I2C_CLR_GEN_CALL is the GEN_CALL interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x0068	I2C_CLR_GEN_CALL	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															clr_gen_call
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RO	clr_gen_call	When this register is read, the GEN_CALL interrupt is cleared.

I2C_ENABLE

I2C_ENABLE is the I^2C enable register. It is used to enable or disable the I^2C.

When the I^2C controller is transferring data, that is, I2C_STATUS[ACTIVITY] is 1, the controller can be disabled. In this case, the following precautions should be taken:

- If the I^2C controller is disabled when it is transmitting data, the controller stops transmitting data after the current byte is transmitted and the data in the TX_FIFO is cleared.
- If the I^2C controller is disabled when it is receiving data, the controller does not respond to this transfer after the current byte is received. That is, the NACK is sent.

Offset Address	Register Name	Total Reset Value
0x006C	I2C_ENABLE	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															enable
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:1]	RO	reserved	Reserved.
[0]	RW	enable	I^2C enable register. 0: The I^2C is disabled. 1: The I^2C is enabled.

I2C_STATUS

I^2C_STATUS is the I^2C status register.

Offset Address	Register Name	Total Reset Value
0x0070	I2C_STATUS	0x0006

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved											rff	rfne	tfe	tfnf	activity
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bits	Access	Name	Description
[15:5]	RO	reserved	Reserved.
[4]	RO	rff	This bit indicates whether the RX_FIFO is full. 0: The RX_FIFO is not full. 1: The RX_FIFO is full.
[3]	RO	rfne	This bit indicates whether the RX_FIFO is empty. 0: The RX_FIFO is empty. 1: The RX_FIFO is not empty.
[2]	RO	tfe	This bit indicates whether the TX_FIFO is empty. 0: The TX_FIFO is not empty. 1: The TX_FIFO is empty.
[1]	RO	tfnf	This bit indicates whether the TX_FIFO is full. 0: The TX_FIFO is full. 1: The TX_FIFO is not full.
[0]	RO	activity	I^2C bus status. 0: The I^2C bus is idle. 1: The I^2C bus is busy.

I2C_TXFLR

I2C_TXFLR is the TX_FIFO data count indication register.

Offset Address	Register Name	Total Reset Value
0x0074	I2C_TXFLR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												txflr			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:4]	RO	reserved	Reserved.

[3:0]	RO	txflr	These bits indicate the count of data segments in TX_FIFO.

I2C_RXFLR

I2C_RXFLR is the RX_FIFO data count indication register.

Offset Address	Register Name	Total Reset Value
0x0078	I2C_RXFLR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												rxflr			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:4]	RO	reserved	Reserved.
[3:0]	RO	rxflr	These bits indicate the count of data segments in RX_FIFO.

I2C_TX_ABRT_SOURCE

I2C_TX_ABRT_SOURCE is the TX_ABRT source interrupt register.

Offset Address	Register Name	Total Reset Value
0x0080	I2C_TX_ABRT_SOURCE	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				arb_master_dis	abrt_10b_rd_norstrt	abrt_sbyte_norstrt	abrt_hs_norstrt	abrt_sbyte_ackdet	abrt_hs_ackdet	abrt_gcall_read	abrt_gcall_noack	abrt_txdata_noack	abrt_10addr2_noack	abrt_10addr1_noack	abrt_7b_addr_noack
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:12]	RO	reserved	Reserved.
[11]	RO	arb_master_dis	This bit indicates whether an error occurs due to the attempt of a master operation when the master function is disabled. 0: No such error occurs. 1: This error occurs.

[10]	RO	abrt_10b_rd_norstrt	This bit indicates whether an error occurs when the restart function is not supported and the I^2C as a master sends a read command to the 10-bit slave address. 0: No such error occurs. 1: This error occurs.
[9]	RO	abrt_sbyte_norstrt	This bit indicates whether an error occurs when the restart function is not supported and the I^2C as a master attempts to send the start byte. 0: No such error occurs. 1: This error occurs.
[8]	RO	abrt_hs_norstrt	This bit indicates whether an error occurs when the restart function is not supported and the I^2C as a master attempts to perform a high-speed operation. 0: No such error occurs. 1: This error occurs.
[7]	RO	abrt_sbyte_ackdet	This bit indicates whether an error occurs when an ACK signal is received after the I^2C as a master sends the start byte. 0: No such error occurs. 1: This error occurs.
[6]	RO	abrt_hs_ackdet	This bit indicates whether an error occurs when the I^2C as a master attempts to perform a high-speed operation and the high-speed host code is acknowledged. 0: No such error occurs. 1: This error occurs.
[5]	RO	abrt_gcall_read	This bit indicates whether an error occurs when the I^2C as a master sends a general call but CPU sends a read command to the I^2C. 0: No such error occurs. 1: This error occurs.
[4]	RO	abrt_gcall_noack	This bit indicates whether an error occurs when the I^2C as a master sends a general call and no ACK signal is received. 0: No such error occurs. 1: This error occurs.

[3]	RO	abrt_txdata_noack	This bit indicates whether an error occurs because the transmitted address is acknowledged by the slave whereas the transmitted data is not acknowledged when the I^2C acts as a master transmitter. 0: No such error occurs. 1: This error occurs.
[2]	RO	abrt_10addr2_noack	This bit indicates whether an error occurs because the second byte of the 10-bit transmitted address is not acknowledged when the I^2C acts as a master. 0: No such error occurs. 1: This error occurs.
[1]	RO	abrt_10addr1_noack	This bit indicates whether an error occurs because the first byte of the10-bit transmitted address is not acknowledged when the I^2C acts as a master. 0: No such error occurs. 1: This error occurs.
[0]	RO	abrt_7b_addr_noack	This bit indicates whether an error occurs because the 7-bit transmitted address is not acknowledged when the I^2C acts as a master. 0: No such error occurs. 1: This error occurs.

I2C_DMA_CR

I2C_DMA_CR is the I^2C DMA channel enable control register.

Offset Address	Register Name	Total Reset Value
0x0088	I2C_DMA_CR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved														tdmae	rdmae
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:2]	RO	reserved	Reserved.
[1]	RW	tdmae	This bit indicates whether to enable the DMA channel of the TX_FIFO. 0: Do not enable the DMA channel of the TX_FIFO. 1: Enable the DMA channel of the TX_FIFO.

[0]	RW	rdmae	This bit indicates whether to enable the DMA channel of the RX_FIFO. 0: Do not enable the DMA channel of the RX_FIFO. 1: Enable the DMA channel of the RX_FIFO.

I2C_DMA_TDLR

I2C_DMA_TDLR is the TX_FIFO threshold configuration register for DMA operation.

Offset Address	Register Name	Total Reset Value
0x008C	I2C_DMA_TDLR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved													dmatdl		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:3]	RO	reserved	Reserved.
[2:0]	RW	dmatdl	These bits indicate the threshold when the TX_FIFO performs the DMA operation.

I2C_DMA_RDLR

I2C_DMA_RDLR is the RX_FIFO threshold configuration register for DMA operation.

Offset Address	Register Name	Total Reset Value
0x0090	I2C_DMA_RDLR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved													dmardl		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:3]	RO	reserved	Reserved.
[2:0]	RW	dmardl	These bits indicate the threshold when the RX_FIFO performs the DMA operation. The actual value is equal to the configured value plus 1.

11.2 UART

11.2.1 Overview

The universal asynchronous receiver transmitter (UART) interface is an asynchronous serial communications interface. It performs serial-to-parallel conversion on the data received from peripherals and sends the converted data to the internal bus. It also performs parallel-to-serial conversion and then sends the obtained data to peripherals. The UART is mainly used to interconnect with the UART of the external chip so that the two chips can communicate with each other.

The Hi3515 provides the following four UART units:

- UART0

  Mainly used for debugging.
- UART1

  Mainly used for connecting the RS-485 bus and controlling the pan/tilt.
- UART2 and UART3

  Mainly used for extending the interface, such as the external micro controller unit (MCU).

11.2.2 Features

The UART module has the following features:

- Supports16 x 8 bit transmit first-in, first-out (FIFO) and 16 x 12 bit receive FIFO.
- Supports the programmable data bit width and stop bit width. By programming, the data bit can be set to 5 bits, 6 bits, 7 bits, or 8 bits and the stop bit can be set to 1 bit or 2 bits.
- Supports parity check or no check.
- Supports the programmable transfer rate.
- Supports the receive FIFO interrupt, transmit FIFO interrupt, receive timeout interrupt, and error interrupt.
- Supports the query of the raw interrupt status and the masked interrupt status.
- Supports disabling the UART module or the transmit/receive function of the UART through programming, thus reducing power consumption.
- Supports disabling the UART clock, thus reducing power consumption.
- UART0, UART1, and UART2 support the DMA operation, while UART3 does not.

11.2.3 Signal Description

Table 11-7 lists the signals of the UART0 interface.

Table 11-7 Signals of the UART0 interface

Signal Name	Direction	Description	Corresponding Pin
UART0_RXD	I	UART0 receive data signal.	URXD0
UART0_TXD	O	UART0 transmit data signal.	UTXD0

Table 11-8 lists the signals of the UART1 interface.

Table 11-8 Signals of the UART1 interface

Signal Name	Direction	Description	Corresponding Pin
UART1_RXD	I	UART1 receive data signal.	URXD1
UART1_TXD	O	UART1 transmit data signal.	UTXD1
UART1_RTS	O	UART1 request transmit signal.	URTSN1
UART1_CTS	I	UART1 clear transmit signal.	UCTSN1

Table 11-9 lists the signals of the UART2 interface.

Table 11-9 Signals of the UART2 interface

Signal Name	Direction	Description	Corresponding Pin
UART2_RXD	I	UART2 receive data signal, multiplexed with several VI and GPIO signals. For details about the multiplexing configuration information, see section 11.2.5 "Pin Multiplexing."	VI0HS
UART2_TXD	O	UART2 transmit data signal, multiplexed with the VI and GPIO pins. For details about the multiplexing configuration information, see section 11.2.5 "Pin Multiplexing."	VI0VS

Table 11-10 lists the signals of the UART3 interface.

Table 11-10 Signals of the UART3 interface

Signal Name	Direction	Description	Corresponding Pin
UART3_RXD	I	UART3 receive data signal, multiplexed with the VI and GPIO pins. For details about the multiplexing configuration information, see section 11.2.5 "Pin Multiplexing."	VI2HS

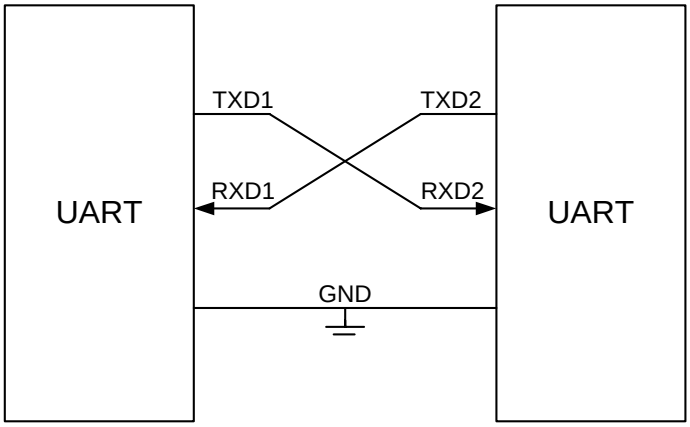
Signal Name	Direction	Description	Corresponding Pin
UART3_TXD	O	UART0 transmit data signal, multiplexed with the VI and GPIO pins. For details about the multiplexing configuration information, see section 11.2.5 "Pin Multiplexing."	VI2VS

11.2.4 Function Description

Application Block Diagram

Figure 11-4 and Figure 11-5 show the typical applications of the UART.

Figure 11-4 Typical application block diagram 1 of the UART

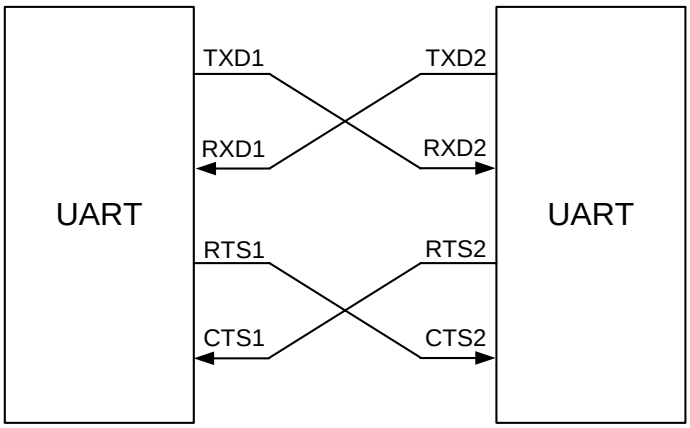


The UART serves as an asynchronous bidirectional serial interface. Through the UARTs connected by two data lines, a simplified and effective data transfer mode is implemented. UART0, UART1, UART2, and UART3 support this data transfer mode.

CAUTION

Under the request-to-send (RTS) flow control, the output values of the RTS cannot be specified through the UART_CR register.

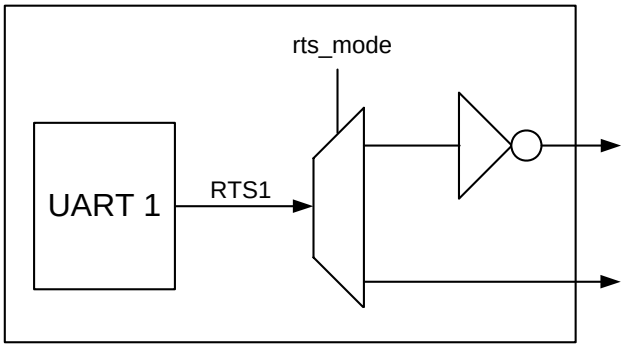
Figure 11-5 Typical application block diagram 2 of the UART



If the interconnected chip requires to control stream through RTS or clear-to-send (CTS), the RTS pin and CTS pin can be connected to control data transmit and receive through their handshakes. When UART1 is connected to the RS-485 bus chip, the RTS pin and CTS pin need to be connected.

Figure 11-6 shows the output mode of the RTS signal.

Figure 11-6 Block diagram of the RTS signal output mode

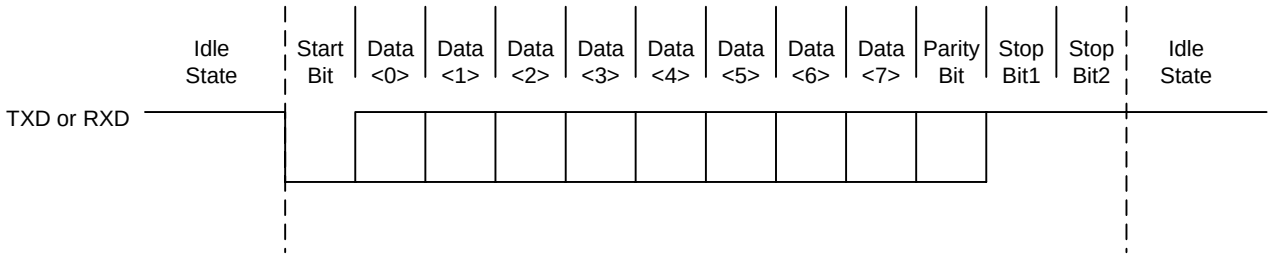


Under the RTS flow control, the output values of the RTS cannot be configured through the UART_CR register. The level mode of the RTS output, however, can be controlled by configuring SC_PERCTRL11[uart1_rtsmode] of the system controller.

Function Principle

A frame transfer of the UART involves the start signal, data signal, parity bit, and stop signal, as shown in Figure 11-7. The data frame is output from the TXD port of one UART and is input to the RXD port of another UART.

Figure 11-7 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 as required. For example, it can be set to 5 bits, 6 bits, 7 bits, or 8 bits.

- 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. In addition, 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 the data frame. The stop bit can be configured as 1 bit or 2 bits. The high level of TXD indicates the end of the data frame.

11.2.5 Operating Mode

Pin Multiplexing

Before using UART2 and UART3, you can enable the UART function by configuring the registers reg0–reg1 and reg10–reg11.

Clock Gating

When the current data transfer is complete and a new data transfer does not start, the UART clock can be disabled if hardware is idle (that is, UART_FR bit[3] = 0). To disable the UART clock, do as follows:

Step 1 Read UART_FR.

Step 2 If UART_FR bit[3] is 0, set UART_CR[uarten] to 0, and then go to Step 3. If UART_FR bit[3] is 1, wait and go to Step 1.

Step 3 Set SC_PERDIS bit[7:4] to 0xF to disable the UART clock.

----End

Baud Rate Configuration

The baud rate of the UART is configured through UART_IBRD and UART_FBRD and can be calculated by the following formula: Current baud rate = Reference clock frequency of the UART/(16 x clock divider).

The clock divider consists of the integer part and the decimal part that correspond to UART_IBRD and UART_FBRD respectively.

For example, assume that the frequency of the internal bus clock is 60 MHz. If UART_IBRD is set to 0x1E and UART_FBRD is set to 0x00, then the current baud rate is 60/(16 x 30) = 0.125 Mbit/s.

The typical values of the UART baud rate are 9,600 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 configure clock divider register:

If the required baud rate is 230,400 bit/s and the frequency of UARTCLK is 100 MHz, then the clock divider is $(100 \times 10^6)/(16 \times 230{,}400) = 27.1267$. The integer part IBRD is 27 and the decimal part FBRD is 0.1267.

The value of the 6-bit UART_FBRD is calculated by following the formula: m = integer(FBRD x 2^n + 0.5), where n indicates the data width of UART_FBRD. In this case, m = integer(0.1267 x 2^6 + 0.5) = 8. Then, 0x001B is configured in UART_IBRD, whereas 0x08 is configured in UART_FBRD.

When the decimal part of the clock divider is set to 8, the actual value of the baud rate divider is 27 + 8/64 = 27.125, and thus the baud rate is $(100 \times 10^6)/(16 \times 27.125) = 230414.75$ and the error rate is (230414.75 – 230400)/230400 x 100 = 0.006%

The maximum error rate is 1/64 x 100 = 1.56% when the 6-bit UART_FBRD is used. If m = 1, the accumulated error rate is greater than 64 clock periods.

Soft Reset

The UART controller can be independently reset by configuring the system controller.

- The UART0 controller is independently reset by setting SC_PERCTRL8 bit[6] to 1.
- The UART1 controller is independently reset by setting SC_PERCTRL8 bit[7] to 1.
- The UART2 controller is independently reset by setting SC_PERCTRL8 bit[8] to 1.
- The UART3 controller is independently reset by setting SC_PERCTRL8 bit[9] to 1.

After the reset, the configuration registers are restored to default values. Therefore, these registers must be initialized again.

Data Transfer in Interrupt or Query Mode

To initialize the UART, do as follows:

Step 1 Write 0 to UART_CR bit[0] to disable the UART.

Step 2 Write corresponding values 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 transmit and receive FIFO thresholds.

Step 5 If the driver runs in interrupt mode, set UART_IMSC to enable the corresponding interrupt. If in query mode, disable the corresponding interrupt.

Step 6 Write 1 to UART_CR bit[0] to enable the UART. The initialization process is complete.

----End

To transmit data, do as follows:

Step 1 Write the data to be transmitted to UART_DR and start data transmit.

Step 2 In query mode, detect the TX_FIFO state by reading UART_FR bit[5] during continuous data transmit and then determine whether to transmit data to the TX_FIFO according to the TX_FIFO state. In interrupt mode, determine whether to transmit data to the TX_FIFO according to the corresponding interrupt status bit.

Step 3 Check whether the UART transmits all data by detecting UART_FR bit[7]. If UART_FR bit[7] is 1, it indicates that the UART transmits all data. Otherwise, the UART does not transmit all data.

----End

To receive data, do as follows:

- In query mode, detect the RX_FIFO state by reading UART_FR bit[4] during data receive and then determine whether to read data from the RX_FIFO according to the RX_FIFO state.
- In interrupt mode, determine whether to read data from the RX_FIFO according to the corresponding interrupt status bit.

Data Transfer in DMA Mode

CAUTION

UART0, UART1 and UART2 support the DMA transfer mode, whereas UART3 does not.

To initialize the UART, do as follows:

Step 1 Write 0 to UART_CR bit[0] to disable the UART.

Step 2 Write corresponding values 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 transmit and receive FIFO thresholds.

Step 5 If the driver runs in interrupt mode, set UART_IMSC to enable the corresponding interrupt. If in query mode, disable the corresponding interrupt.

Step 6 Write 1 to UART_CR bit[0] to enable the UART. The initialization process is complete.

----End

To transmit data, do as follows:

Step 7 Configure the DMA data channel, including the source and destination addresses, number of data segments to be transmitted, and transfer type. For details, see the description in section 3.5 "DMAC."

Step 8 Set UART_DMACR to 0x2 to enable the DMA transmit function of the UART.

Step 9 Check whether the data is transmitted completely according to the DMA interrupt. If all data is transmitted, disable the DMA transmit function of the UART.

----End

To receive data, do as follows:

Step 10 Configure the DMA data channel, including data transfer source and destination addresses, data receive area address, number of data segments to be transmitted, and transfer type.

Step 11 Set UART_DMACR to 0x1 to enable the DMA receive function of the UART.

Step 12 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

11.2.6 Register Summary

The Hi3515 provides four UARTs, namely, UART0, UART1, UART2, and UART3. Their base addresses are as follows:

- Base address of UART0: 0x2009_0000
- Base address of UART1: 0x200A_0000
- Base address of UART2: 0x200B_0000
- Base address of UART3: 0x200C_0000

Table 11-11 lists the UART registers.

Table 11-11 Summary of the UART registers

Offset Address	Register	Description	Page
0x000	UART_DR	Data register	11-39
0x004	UART_RSR	Receive status register or error clear register	11-40
0x008–0x014	RESERVED	Reserved	-
0x018	UART_FR	Flag register	11-41
0x01C–0x020	RESERVED	Reserved	-
0x024	UART_IBRD	Integer baud rate register	11-42
0x028	UART_FBRD	Decimal baud rate register	11-43
0x02C	UART_LCR_H	Line control register	11-43
0x030	UART_CR	Control register	11-45

Offset Address	Register	Description	Page
0x034	UART_IFLS	Interrupt FIFO threshold select register	11-46
0x038	UART_IMSC	Interrupt mask register	11-47
0x03C	UART_RIS	Raw interrupt status register	11-48
0x040	UART_MIS	Masked interrupt status register	11-49
0x044	UART_ICR	Interrupt clear register	11-50
0x048	UART_DMACR	DMA control register	11-51

11.2.7 Register Description

UART_DR

UART_DR is the UART data register which stores the received and transmitted data. The receive status can be queried by reading this register.

Offset Address	Register Name	Total Reset Value
0x000	UART_DR	0x00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				oe	be	pe	fe	data							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:12]	RO	reserved	Reserved.
[11]	RO	oe	Overrun error. 0: No overrun error occurs. 1: An overrun error occurs. That is, a segment of data is received when the receive FIFO is full.
[10]	RO	be	Break error. 0: No break error occurs. 1: A break error occurs. That is, the receive data input signal keeps low longer than a full word transfer. A full word consists of a start bit, data bit, parity bit, and stop bit.
[9]	RO	pe	Parity error. 0: No parity error occurs. 1: A parity error occurs.
[8]	RO	fe	Frame error.

			0: No frame error occurs. 1: A frame error (namely, stop bit error) occurs.
[7:0]	RW	data	Data to be transmitted or received.

UART_RSR

UART_RSR is the receive status register or error clear register.

- It acts as the receive status register when read.
- It acts as the error clear register when written.

The receive status can be queried from the UART_DR register. The break, frame, and parity status information read from UART_DR takes priority over that read from UART_RSR. That is, the status information in UART_DR is updated faster than that in UART_RSR.

UART_RSR is reset when any value is written to it.

Offset Address	Register Name	Total Reset Value
0x004	UART_RSR	0x00

Bit	7	6	5	4	3	2	1	0
Name	reserved				oe	be	pe	fe

Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:4]	RO	reserved	Reserved.
[3]	RW	oe	Overrun error. 0: No overrun error occurs. 1: An overrun error occurs. When the FIFO is full, the following data cannot be written to the FIFO and only the shift register is overrun. Therefore, the contents at the FIFO are valid. In this case, the CPU must read the data immediately to spare buffer space.
[2]	RW	be	Break error. 0: No break error occurs. 1: A break error occurs. Condition for the break error: The receive data input signal keeps low longer than a full word transfer. A full word consists of a start bit, data bit, parity bit, and stop bit.
[1]	RW	pe	Parity error. 0: No parity error occurs. 1: A parity error of the received data occurs. In FIFO mode, the error is associated with the data at the top of the FIFO.
[0]	RW	fe	Frame error. 0: No frame error occurs. 1: An error occurs at the stop bit of the received data. The valid stop bit is 1.

UART_FR

UART_FR is the UART flag register.

Offset Address	Register Name	Total Reset Value
0x018	UART_FR	0x0012

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								txfe	rxff	txff	rxfe	busy	reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bits	Access	Name	Description
[15:8]	RO	reserved	Reserved.

[7]	RO	txfe	The meaning of the bit is determined by UART_LCR_H[fen]. If UART_LCR_H[fen] is 0, set the bit to 1 when the transmit holding register is empty. If UART_LCR_H[fen] is 1, set the bit to 1 when the transmit FIFO is empty.
[6]	RO	rxff	The meaning of the bit is determined by UART_LCR_H[FEN]. If UART_LCR_H[fen] is 0, set the bit to 1 when the receive holding register is full. If UART_LCR_H[fen] is 1, set the bit to 1 when the receive FIFO is full.
[5]	RO	txff	The meaning of the bit is determined by UART_LCR_H[FEN]. If UART_LCR_H[fen] is 0, set the bit to 1 when the transmit holding register is full. If UART_LCR_H[fen] is 1, set this bit to 1 when the transmit FIFO is full.
[4]	RO	rxfe	The meaning of the bit is determined by UART_LCR_H[FEN]. If UART_LCR_H[fen] is 0, set the bit to 1 when the receive holding register is empty. If UART_LCR_H[fen] is 1, set the bit to 1 when the receive FIFO is empty.
[3]	RO	busy	UART busy state bit. 0: The UART is idle or data transmit is complete. 1: The UART is busy in transmitting data. If the bit is set to 1, the state is kept until the entire byte (including all bits of the stop bit) is transmitted from the shift register. Regardless of whether the UART is enabled, this bit is set to 1 when the transmit FIFO is not empty.
[2:0]	RO	reserved	Reserved.

UART_IBRD

UART_IBRD is the integer baud rate register.

Offset Address	Register Name	Total Reset Value
0x024	UART_IBRD	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	baud divint															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:0]	RW	baud divint	Integer-baud-rate divider. The bits are cleared after reset.

UART_FBRD

UART_FBRD is the decimal baud rate register.

CAUTION

- The values of the integer and decimal baud rate registers can be updated until the current data is transmitted and received completely.
- The minimum clock divider is 1 and the maximum divider is 65,535 (216 – 1). That is, UART_IBRD cannot be 0; otherwise, UART_FBRD is ignored. Similarly, if UART_IBRD is equal to 65,535 (0xFFFF), UART_IBRD must be 0. If UART_IBRD is greater than 0, the data transmit and receive may fail.
- Assume that UART_FBRD is set to 0x1E and UART_IBRD is set to 0x01. Then, the integer part of the clock divider is 30, the decimal part is 0.015625, and the clock divider is 30.015625.
- Baud rate of the UART = Internal bus frequency/(16 x clock divider) = Internal bus frequency/(16 x 30.015625).

Offset Address	Register Name	Total Reset Value
0x028	UART_FBRD	0x00

Bit	7	6	5	4	3	2	1	0
Name	reserved		baud divfrac					
Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:6]	RO	Reserved	Reserved.
[5:0]	RW	band divfrac	Decimal-baud-rate divider. The bits are cleared after reset.

UART_LCR_H

UART_LCR_H is the 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	Register Name	Total Reset Value
0x02C	UART_LCR_H	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								sps	wlen		fen	stp2	eps	pen	brk
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:8]	RO	reserved	Reserved.
[7]	RW	sps	Parity select. When bit[1], bit[2], and bit[7] of this register are set to 1, the parity bit is 0. When bit[1] and bit[7] are set to 1 and bit[2] is 0, the parity bit is 1. When bit[1], bit[2], and bit[7] are cleared, the stick parity bit is disabled.
[6:5]	RW	wlen	Count of bits in a transmitted or received frame. 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits
[4]	RW	fen	Transmit and receive FIFO enable control. 0: The transmit and receive FIFOs are disabled. 1: The transmit and receive FIFOs are enabled.
[3]	RW	stp2	2-bit stop bit at the end of the transmitted frame. 0: There is no 2-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. The receive logic does not check for the 2-bit stop bit during data receive.
[2]	RW	eps	Parity select in data transmit and receive. 0: The odd parity is generated or performed during data transmit and receive. 1: The even parity is generated or performed during data transmit and receive. When UART_LCR_H[fen] is 0, this bit becomes invalid.
[1]	RW	pen	Parity enable bit. 0: The parity is disabled. 1: The parity is generated at the transmit side and performed at the receive side.

[0]	RW	brk	Break command send. 0: Invalid. 1: After the current data transmit is complete, the UTXD outputs low level continuously. Note: This bit must be kept 1 longer than two complete frames so that the break command runs properly. In normal use, the bit must be set to 0.

UART_CR

UART_CR is the UART control register.

To configure the UART_CR register, do as follows:

Step 1 Write 0 to UART_CR bit[0] to disable the UART.

Step 2 Wait until the current data transmit or receive is complete.

Step 3 Set UART_LCR_H[fen] to 0.

Step 4 Configure UART_CR.

Step 5 Write 1 to UART_CR bit[0] to enable the UART.

----End

Offset Address	Register Name	Total Reset Value
0x030	UART_CR	0x0300

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ctsen	rtsen	reserved		rts	dtr	rxe	txe	lbe	reserved						uarten
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15]	RW	ctsen	CTS hardware flow control enable. 0: CTS hardware flow control is disabled. 1:CTS hardware flow control is enabled. The data can be transmitted only when the nUARTCTS signal is valid.
[14]	RW	rtsen	RTS hardware flow control enable. 0: RTS hardware flow control is disabled. 1: RTS hardware flow control is enabled. The data receive request can be sent only when the receive FIFO has free space.
[13:12]	RO	reserved	Reserved.

[11]	RW	rts	Request send. This bit is inverted from the UART modem status output signal nUARTRTS. 0: The output signal remains unchanged. 1: When this bit is set to 1, the output signal is 0.
[10]	RW	dtr	Data transmit ready. This bit is inverted from the UART modem status output signal nUARTDTR. 0: The output signal remains unchanged. 1: When this bit is set to 1, the output signal is 0.
[9]	RW	rxe	UART receive enable. 0: UART receive is disabled. 1: UART receive is enabled. If the UART is disabled during data receive, the current data receive is stopped.
[8]	RW	txe	UART transmit enable. 0: UART transmit is disabled. 1: UART transmit is enabled. If the UART is disabled during data transmit, the current data transmit is stopped.
[7]	RW	lbe	Loopback enable. 0: Loopback is disabled. 1: The UARTTXD output is looped back to UARTRXD.
[6:1]	RO	reserved	Reserved.
[0]	RW	uarten	UART enable. 0: The UART is disabled. 1: The UART is enabled. If the UART is disabled during data receive and transmit, the data transfer is stopped.

UART_IFLS

UART_IFLS is the interrupt FIFO threshold select register that is used to configure the thresholds. When any of the thresholds are exceeded, the FIFO interrupt (UART_TXINTR or UART_RXINTR) can be triggered.

Offset Address	Register Name	Total Reset Value
0x034	UART_IFLS	0x0012

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	reserved										rxiflsel			txiflsel		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bits	Access	Name	Description
[15:6]	RO	reserved	Reserved.
[5:3]	RW	rxiflsel	Receive interrupt FIFO threshold select. A receive interrupt can be triggered when any of the following conditions is met: 000: receive FIFO ≥ 1/8 full 001: receive FIFO ≥ 1/4 full 010: receive FIFO ≥ 1/2 full 011: receive FIFO ≥ 3/4 full 100: receive FIFO ≥ 7/8 full 101–111: reserved
[2:0]	RW	txiflsel	Transmit interrupt FIFO threshold select. A transmit interrupt can be triggered when any of the following conditions is met: 000: transmit FIFO ≤ 1/8 full 001: transmit FIFO ≤ 1/4 full 011: transmit FIFO ≤ 3/4 full 010: transmit FIFO ≤ 1/2 full 100: transmit FIFO ≤ 7/8 full 101–111: reserved

UART_IMSC

UART_IMSC is the interrupt mask register that is used to mask interrupts.

Offset Address	Register Name	Total Reset Value
0x038	UART_IMSC	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					oeim	beim	peim	feim	rtim	txim	rxim	reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:11]	RO	reserved	Reserved.
[10]	RW	oeim	Overrun error interrupt mask status. 0: The interrupt is masked. 1: The interrupt is not masked.

[9]	RW	beim	Break error interrupt mask status. 0: The interrupt is masked. 1: The interrupt is not masked.
[8]	RW	peim	Parity interrupt mask status. 0: The interrupt is masked. 1: The interrupt is not masked.
[7]	RW	feim	Frame error interrupt mask status. 0: The interrupt is masked. 1: The interrupt is not masked.
[6]	RW	rtim	Receive timeout interrupt mask status. 0: The interrupt is masked. 1: The interrupt is not masked.
[5]	RW	txim	Transmit interrupt mask status. 0: The interrupt is masked. 1: The interrupt is not masked.
[4]	RW	rxim	Receive interrupt mask status. 0: The interrupt is masked. 1: The interrupt is not masked.
[3:0]	RO	reserved	Reserved.

UART_RIS

UART_RIS is the raw interrupt status register. The contents of this register are not affected by the UART_IMSC register.

Offset Address	Register Name	Total Reset Value
0x03C	UART_RIS	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					oeris	beris	peris	feris	rtris	txris	rxris	reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:11]	RO	reserved	Reserved.
[10]	RO	oeris	Raw overrun error interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.

[9]	RO	beris	Raw break error interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	peris	Raw parity interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	feris	Raw error interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[6]	RO	rtris	Raw receive timeout interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	txris	Raw transmit interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	rxris	Raw receive interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[3:0]	RO	reserved	Reserved.

UART_MIS

UART_MIS is the masked interrupt status register. The contents of this register are the result obtained after the raw interrupt status is ANDed with the interrupt mask status.

Offset Address	Register Name	Total Reset Value
0x040	UART_MIS	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					oemis	bemis	pemis	femis	rtmis	txmis	rxmis	reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:11]	RO	reserved	Reserved.
[10]	RO	oemis	Masked overrun error interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.

[9]	RO	bemis	Masked break error interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	pemis	Masked parity interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	femis	Masked error interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[6]	RO	rtmis	Masked receive timeout interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	txmis	Masked transmit interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	rxmis	Masked receive interrupt status. 0: No interrupt is generated. 1: An interrupt is generated.
[3:0]	RO	reserved	Reserved.

UART_ICR

UART_ICR is the interrupt clear register. When 1 is written to it, the corresponding interrupt is cleared. Writing 0 has no effect.

Offset Address	Register Name	Total Reset Value
0x044	UART_ICR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					oeic	beic	peic	feic	rtic	txic	rxic	reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:11]	RO	reserved	Reserved.
[10]	WO	oeic	Overrun error interrupt clear. 0: Invalid. 1: The interrupt is cleared.

[9]	WO	beic	Break error interrupt clear. 0: Invalid. 1: The interrupt is cleared.
[8]	WO	peic	Parity interrupt clear. 0: Invalid. 1: The interrupt is cleared.
[7]	WO	feic	Error interrupt clear. 0: Invalid. 1: The interrupt is cleared.
[6]	WO	rtic	Receive timeout interrupt clear. 0: Invalid. 1: The interrupt is cleared.
[5]	WO	txic	Transmit interrupt clear. 0: Invalid. 1: The interrupt is cleared.
[4]	WO	rxic	Receive interrupt clear. 0: Invalid. 1: The interrupt is cleared.
[3:0]	RO	reserved	Reserved.

UART_DMACR

UART_DMACR is the DMA control register. It is used to enable the DMA function of the transmit and receive FIFOs.

Offset Address	Register Name	Total Reset Value
0x048	UART_DMACR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved													dmaonerr	txdmae	rxdmae
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:3]	RO	Reserved	Reserved.

[2]	RW	Dmaonerr	DMA enable control for the receive channel when the UART error interrupt (UARTEINTR) occurs. 0: When the UARTEINTR is valid, the DMA request output (UARTRXDMASREQ or UARRTXDMABREQ) of the receive channel is valid. 1: When the UARTEINTR is valid, the DMA request output (UARTRXDMASREQ or UARRTXDMABREQ) of the receive channel is invalid.
[1]	RW	txdmae	DMA enable control of the transmit FIFO. 0: The DMA function is disabled. 1: The DMA function is enabled.
[0]	RW	rxdmae	DMA enable control of the receive FIFO. 0: The DMA function is disabled. 1: The DMA function is enabled.

11.3 SPI

11.3.1 Overview

The serial peripheral interface (SPI) controller acts as a master or slave device to communicate with external devices in synchronous serial mode. The SPI supports the following protocols:

- Motorola SPI-compatible interface
- Texas Instruments synchronous serial interface
- National Semiconductor Microwire interface

The SPI is mainly used to connect the touch screen, secure digital (SD) card, and wireless fidelity (WiFi).

11.3.2 Features

The SPI module has the following features:

- Supports the master operating mode where up to two slave devices are allowed.
- Supports the slave operating mode.
- Supports programmable frequency of the interface clock.
- Supports two separate FIFOs, one working as a receive FIFO and the other as a transmit FIFO. Each of them is 16-bit wide and 8-location deep.
- Supports three frame formats: SPI, Microwire, and TI synchronous serial.
- Supports programmable frame size: 4 bits to 16 bits.
- Provides internal loopback test mode.
- Supports the DMA operation.

11.3.3 Signal Description

Table 11-12 lists SPI interface signals.

Table 11-12 SPI interface signals

Signal Name	Direction	Description	Corresponding Pin
SPICK	I/O	SPI clock output, multiplexed with the video output (VO) pin only. For details about the multiplexing configuration, see 11.3.5 "Pin Multiplexing."	VOCK
SPIDI	I	SPI data input, multiplexed with the VO pin only. For details about the multiplexing configuration, see 11.3.5 "Pin Multiplexing."	VODAT0
SPIDO	O	SPI data output, multiplexed with the VO pin only. For details about the multiplexing configuration, see 11.3.5 "Pin Multiplexing."	VODAT1
SPICSN0 /SPICSN1	I/O	When the SPI is configured in the SPI or Microwire frame format, the signal is a chip select (CS) signal. When the SPI is configured in the TI frame format, the signal is used as a frame sync signal. For details about the multiplexing configuration, see 11.3.5 "Pin Multiplexing."	VODAT2 /VODAT3

11.3.4 Function Description

Application Block Diagram

Figure 11-8 shows the application block diagram when the SPI is connected to a single slave device. When the SPI is connected to a signal slave device, the default chip select pin SPICSN0 is used.

Figure 11-8 Application block diagram when the SPI is connected to a single slave device

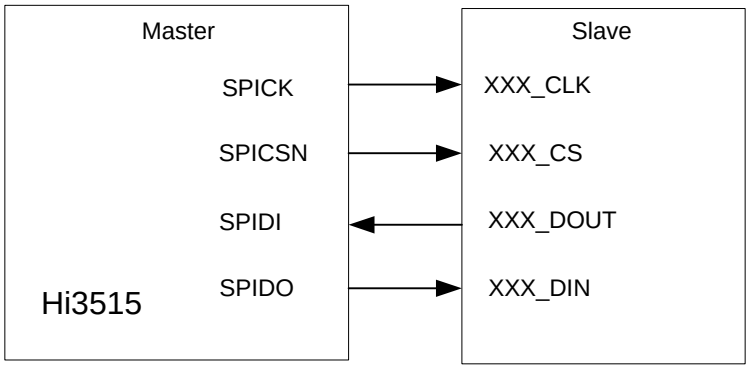


Figure 11-9 shows the application block diagram when the SPI is connected to two slave devices. When the SPI is connected to two slave devices externally, the chip select signals carried by the SPI are not sufficient. In this case, the chip signals are sent to either SPICSN0 or SPICSN1 by configuring the system controller SC_PERCTRL11[spi_port]. The two pins cannot be valid at the same time. When SC_PERCTRL11[spi_port]] is set to 0, SPICS0N is valid. When SC_PERCTRL11[spi_port] is set to 1, SPICS1N is valid.

Figure 11-9 Application block diagram when the SPI is connected to two slave devices

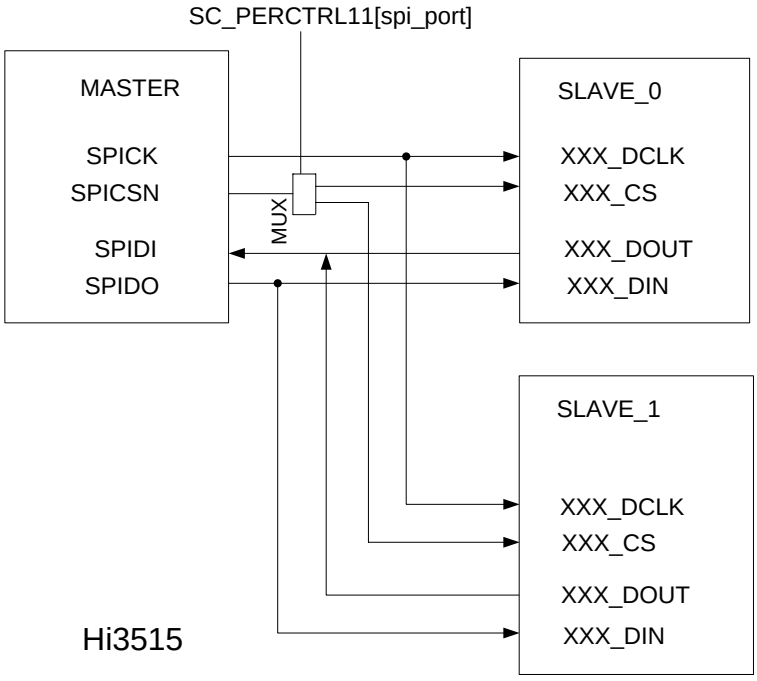
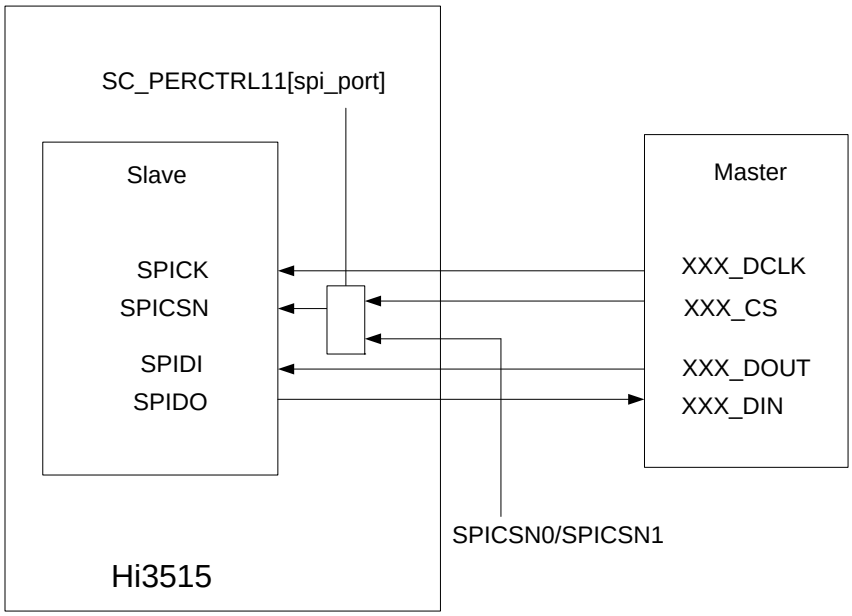


Figure 11-10 shows the application block diagram when the SPI acts as a slave device. The external master device selects either SPICSN0 or SPICSN1 as the input pin. In this case, ensure that SC_PERCTRL11[spi_port] is configured properly. Otherwise, the chip select signal of the external master device cannot be input to the SPI interface correctly. It is recommended to use the default chip select interface, namely, SPICSN0.

Figure 11-10 Application block diagram of the SPI acting as a slave device



Function Principle

NOTE

The symbol spo indicates the polarity of SPICLKOUT and the symbol sph indicates the phase of SPICLKOUT.

From Figure 11-11 to Figure 11-22, the related acronyms and convention are as follows:

- MSB: most significant bit
- LSB: least significant bit
- Q: undefined signal

Figure 11-11 shows the Motorola SPI single frame format when both spo and sph are equal to 0.

Figure 11-11 Motorola SPI single frame format (spo = 0, sph = 0)

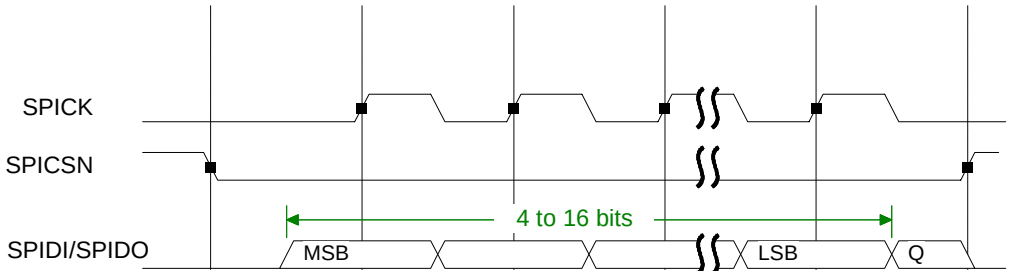
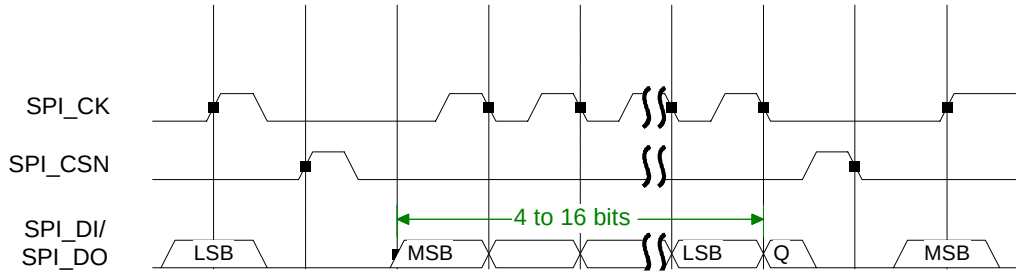


Figure 11-12 shows the Motorola SPI continuous frame format when both spo and sph are equal to 0.

Figure 11-12 Motorola SPI continuous frame format (spo = 0, sph = 0)



When the SPI is idle in this mode:

- The SPICK signal is set to low.
- The SPICSN signal is set to high.
- The transmit data line SPIDO is forced to low.

When the SPI is enabled and valid data is ready in the transmit FIFO, setting the SPICSN signal to low enables a data transfer to start. After the data transfer starts, the enabled slave data is placed on the input line SPIDI of the master device. Half SPICK clock period later, the valid master data is transmitted to the SPIDO pin. At this time, both the master and slave data become valid. The SPICK master clock pin changes to high in the coming half SPICK clock period. Then, data is captured on the rising edge and is transmitted on the falling edge of the SPICK clock.

If a single word is to be transferred, the SPICSN signal is restored to high one SPICK clock period later after the last bit is captured.

For a continuous transfer, the SPICSN signal must be pulled up for one SPICK clock period between the transfers of two words. When sph is 0, the slave select pin fixes the data of the internal serial device register to maintain the data unchanged. Therefore, the SPICSN signal must be pulled up between the transfers of two words in continuous transfer mode. When the continuous transfer ends, the SPICSN is restored to high one SPICK clock period later after the last bit is captured.

Figure 11-13 shows the Motorola SPI single frame format when spo is equal to 0 and sph is equal to 1.

Figure 11-13 Motorola SPI single frame format (spo = 0, sph = 1)

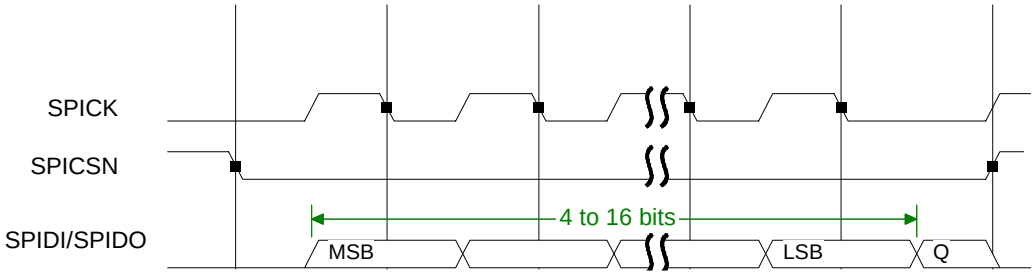
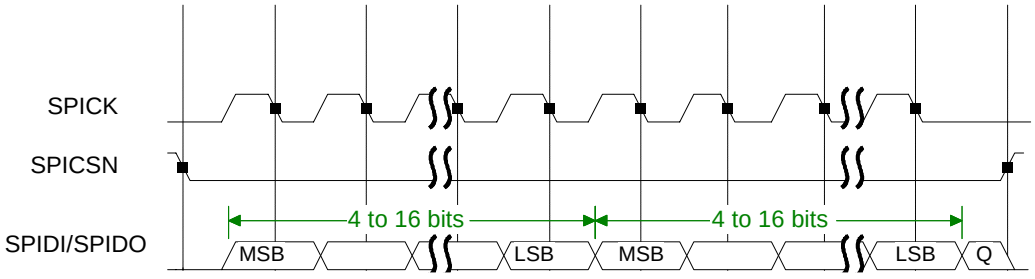


Figure 11-14 shows the Motorola SPI continuous frame format when spo is equal to 0 and sph is equal to 1.

Figure 11-14 Motorola SPI continuous frame format (spo = 0, sph = 1)



When the SPI is idle in this mode:

- The SPICK signal is set to low.
- The SPICSN is set to high.
- The transmit data line SPIDO is forced to low.

When the SPI is enabled and valid data is ready in the transmit FIFO, setting the SPICSN signal to low enables a data transfer to start. If the data transfer starts, the master and slave data become valid on their respective transfer line half SPICK clock period later. Meanwhile, SPICK becomes valid on a rising edge. Then, data is captured on the rising edge and is transmitted on the falling edge of the SPICK clock.

If a single word is to be transferred, the SPICSN signal is restored to high one SPICK clock period later after the last bit is captured.

For a continuous transfer, SPICSN remains low between the two word transfers. After the continuous transfer ends, SPICSN is restored to high one SPICK clock period later after the last bit is captured.

Figure 11-15 shows the Motorola SPI single frame format when spo is equal to 1 and sph is equal to 0.

Figure 11-15 Motorola SPI single frame format (spo = 1, sph = 0)

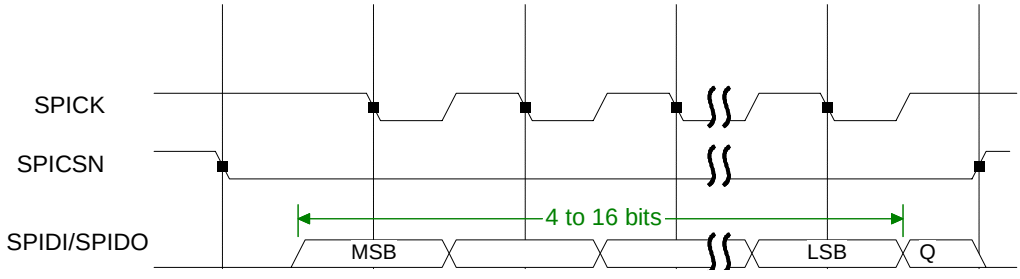
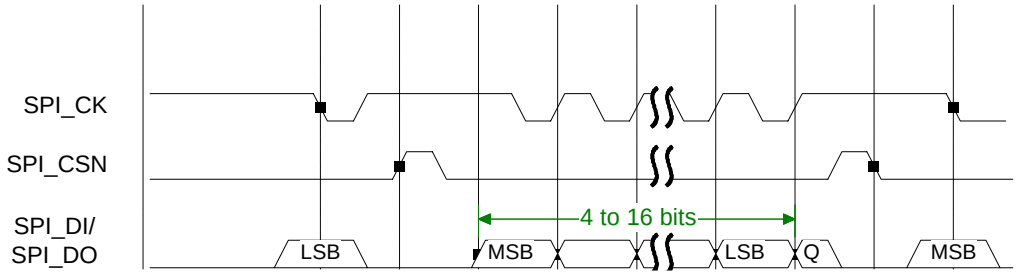


Figure 11-16 shows the Motorola SPI continuous frame format when spo is equal to 1 and sph is equal to 0.

Figure 11-16 Motorola SPI continuous frame format (spo = 1, sph = 0)



When the SPI is idle in this mode:

- The SPICK signal is set to high.
- The SPICSN signal is set to high.
- The transmit data line SPIDO is forced to low.

When the SPI is enabled and valid data is ready in the transmit FIFO, setting the SPICSN signal to low enables a data transfer to start. After the data transfer starts, the slave data is immediately transmitted to the receive data line SPIDI of the master device. Half SPICK clock period later, the valid master data is transmitted to the SPIDO line. After another half SPICK clock period, the SPICK master pin is set to low. This indicates that the data is captured on the falling edge and transmitted on the rising edge of the SPICK clock.

If a single word is to be transferred, the SPICSN signal is restored to high one SPICK clock period later after the last bit is captured.

For a continuous transfer, SPICSN signal must be pulled up between the transfers of two words. The reason is that when sph is 0, the slave select pin fixes the data of the internal serial device register to maintain the data unchanged. The SPICSN is restored to high one SPICK clock period later after the last bit is captured.

Figure 11-17 shows the Motorola SPI single frame format when both spo and sph are equal to 1.

Figure 11-17 Motorola SPI single frame format (spo = 1, sph = 1)

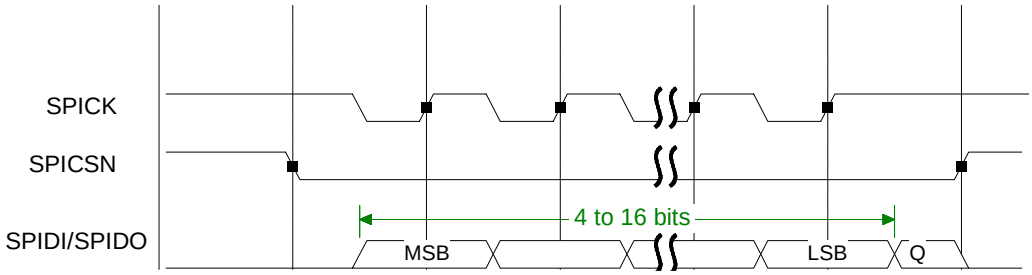
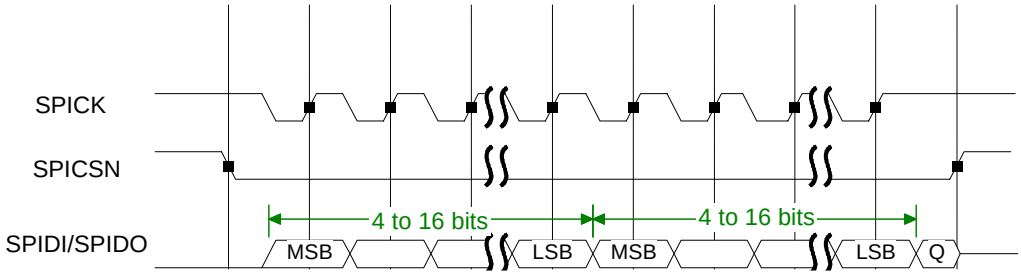


Figure 11-18 shows the Motorola SPI continuous frame format when both spo and sph are equal to 1.

Figure 11-18 Motorola SPI continuous frame format (spo = 1, sph = 1)



When the SPI is idle in this mode:

- The SPICK signal is set to high.
- The SPICSN signal is set to high.
- The transmit data line SPIDO is forced to low.

When the SPI is enabled and valid data is ready in the transmit FIFO, setting the SPICSN master signal to low enables a data transfer to start. If the data transfer starts, the master data and slave data become valid on their respective transfer line after half SPICK clock period. Meanwhile, SPICK becomes valid on a falling edge. Then, the data is captured on the falling edge and transmitted on the rising edge of the SPICK clock.

If a single word is to be transmitted, SPICSN is restored to high one SPICK clock period later after the last bit is captured.

For a continuous transfer, the SPICSN signal remains low. When the transfer ends, the SPICSN signal is restored to high (idle) one SPICK clock period later after the last bit is captured. The end mode for a continuous transfer is the same as that for a single frame transfer.

Figure 11-19 shows the TI synchronous serial single frame format.

Figure 11-19 TI synchronous serial single frame format

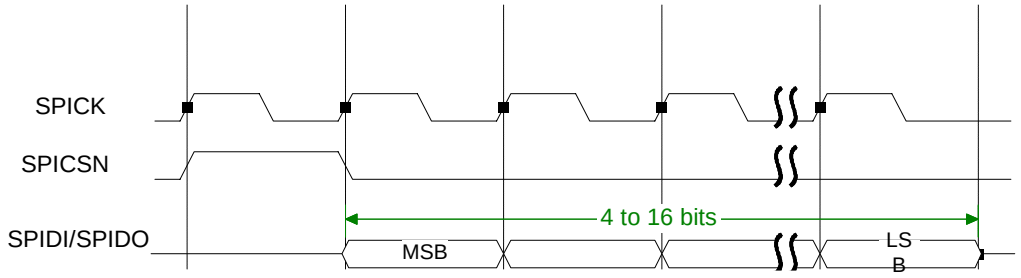
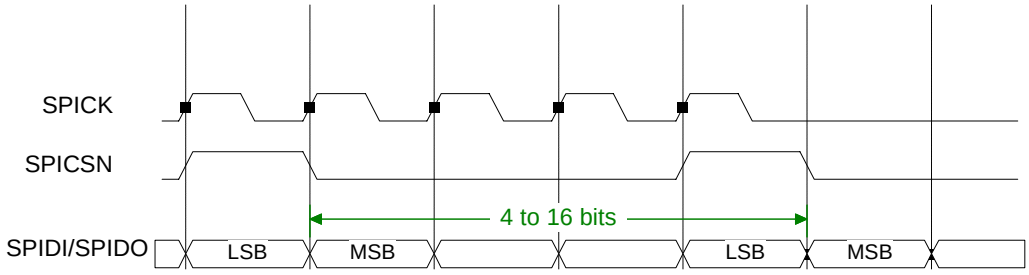


Figure 11-20 shows the TI synchronous serial continuous frame format.

Figure 11-20 TI synchronous serial continuous frame format



When the SPI is idle in this mode:

- The SPICK signal is set to low.
- The SPICSN signal is set to low.
- The transfer data line SPIDO maintains a high impedance.

If there is data in the transmit FIFO, SPICSN generates a high level pulse in one SPICK clock period. Then, the data to be transmitted is transferred from the transmit FIFO to the serial shift register of the transmit logic. The MSBs of the data frames from 4 bits to 16 bits are shift-output from SPIDO on the next rising edge of the SPICX clock. Likewise, the MSBs of the data received from the external serial slave device are shift-input from the SPIDI pin.

The SPI and off-chip serial device stores the data in the serial shift register on the falling edge of the SPICK clock. The receive serial register transmits the data to the receive FIFO on the rising edge of the first SPICK clock after receiving the LSB of data frame.

Figure 11-21 shows the National Semiconductor Microwire single frame format.

Figure 11-21 National Semiconductor Microwire single frame format

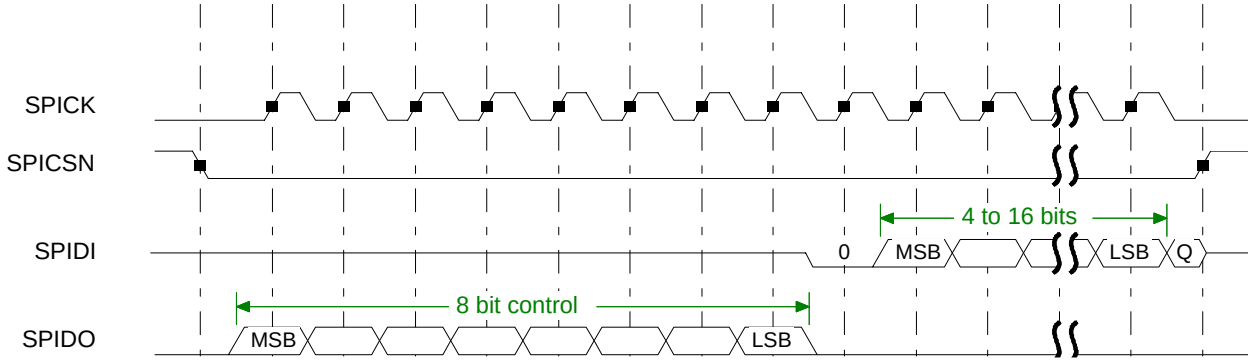
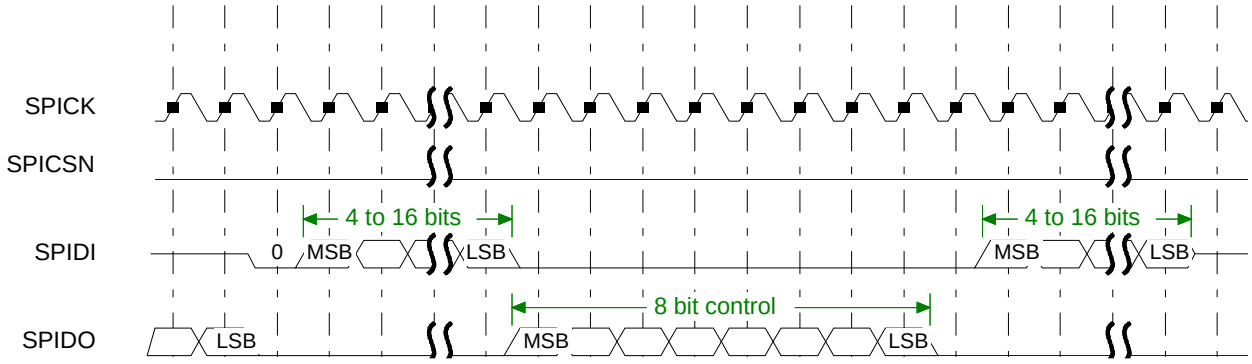


Figure 11-22 shows the National Semiconductor Microwire continuous frame format.

Figure 11-22 National Semiconductor Microwire continuous frame format



The Microwire format is similar to the SPI format because both of them use the transfer technology of master-slave information. The only difference is that the SPI works in full-duplex mode and the Microwire works in half-duplex mode. When 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 the transfer is complete, the off-chip decodes the received data. One clock period after the 8-bit control information is transmitted, the slave device starts to acknowledge the required data. The returned data length is 4 bits to 16 bits, and thus the length of the whole frame is 13 bits to 25 bits.

When the SPI is idle in this mode:

- The SPICK signal is set to low.
- The SPICSN signal is set to high.
- The transmit data line SPIDO is forced to low.

Writing one control byte to the transmit FIFO enables a data transfer to start. The data transfer is triggered on a falling edge of the SPICSN. The data of the transmit FIFO is sent to the serial shift register. The MSBs of the 8-bit control frame are sent to the SPIDO pin. During frame transfer, the SPICSN remains low, whereas the SPIDI maintains a high impedance.

The off-chip serial slave device latches the data to the serial shift register on each rising edge of the SPICK clock. After the slave device latches the last bit, it decodes the received data in the following wait time of one clock period. After that, the slave device sends the requested data to the SPI. Each bit is written to the SPIDI on the falling edge of the SPICK clock. For a single data transfer, the SPICSN is pulled up at the end of the frame one clock period later after the last bit is written to the receive serial register. In this way, the received data is transmitted to the receive 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, the signal SPICSN remains 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 taken from the receive offset register on the falling edge of the SPICK clock.

11.3.5 Operating Mode

Pin Multiplexing

The SPI pins are multiplexed with the VO pin. Therefore, before using SPI pins, you must configure the IO config register to enable the SPI function of the corresponding pins. For details, see the configuration description of the IO Config registers reg28 to reg32.

Clock Gating

When the software completes the current data transfer but does not start a new data transfer, the SPI clock can be disabled if the hardware is in the idle state, that is, SPI_SR[bsy] = 0.

To disable the SPI working clock, do as follows:

Step 1 Read SPI_SR.

Step 2 If SPI_SR[bsy] is 0, write 0 to SPI_CR1[sse] and go to Step 3. If SPI_SR[bsy] is 1, wait and then go to step Step 1.

Step 3 Write SC_PERDIS[spi_clkdis] to disable the SPI working clock.

----End

Clock Configuration

- The minimum SPI working clock frequency is calculated by the following formula:

  SPI working clock frequency (min) ≥ 2 x F_{SPICK} (max)

  When the SPI working clock frequency is preset, the maximum SPI output clock frequency is equal to SPI working clock frequency divided by 2.
- The maximum SPI working clock frequency is calculated by the following formula:

  SPI working clock frequency (max) ≤ 254 x 256 x F_{SPICK} (min)

  When the SPI working clock frequency is preset, the minimum SPI output clock frequency is equal to the SPI working clock frequency divided by 254 x 256.
- The output clock frequency SPICK is calculated by the following formula:

  F_{SPICK} = SPI working clock frequency/(CPSDVSR x (1 + SCR))

The SPI output clock frequency varies with dividers CPSDVSR and SCR. For details, see SPI_CR0 and SPI_CPSR.

Table 11-13 lists the typical configuration of the SPI clock dividers.

Table 11-13 Typical configuration of the SPI clock dividers

SPI Working Clock Frequency (MHz)	CPSDVSR	SCR	SPICK (MHz)
100	2	1	25
100	2	4	10
24	2	1	6
24	2	4	2.4

Soft Reset

The SPI can be soft reset at any time without impacting on the system if there is no data transferred on the bus. The SPI can be soft reset independently by setting SC_PERCTRL10[spi_hrst] to 1. After reset, each configuration register is reset to its default value. Therefore, these registers must be initialized after being reset.

Data Transfer in Interrupt or Query Mode

The initialization process is as follows:

Step 1 Write 0 to SPI_CR1[sse] to disable the SPI.

Step 2 Write corresponding values to SPI_CR0 to set the parameters such as the frame format and transfer data bit width.

Step 3 Configure SPI_CPSR to set the required clock divider.

Step 4 If the driver runs in interrupt mode, set SPI_INTMASK to enable the corresponding interrupt signals. If in query mode, disable the corresponding interrupt signals.

Step 5 Write 1 to SPI_CR1[sse] to enable the SPI. Then, the initialization configuration is complete.

----End

To query and transmit a specified amount of data (for example, four data segments), do as follows:

Step 1 Configure the SPI_CPSR register to set the pre-divider cpsdvsr.

Step 2 Configure SPI_CR0 to set the serial clock rate, frame formats (SPI, TI, and MW), and frame duration dss.

Step 3 Write 1 to SPI_CR1[see] to enable the SPI.

Step 4 Query the status register SPI_SR to check the status of the FIFO. If SPI_SR[tnf] is 1, it indicates that the transmit FIFO is not full. In this case, write four data segments to SPI_DR.

Step 5 Query the status register SPI_SR to check the status of the FIFO until all data segments are transmitted.

Step 6 Write 0 to SPI_CR1[see] to disable the SPI.

----End

To query and receive a specified amount of data (for example, four data segments), do as follows:

Step 7 Configure SPI_CPSR[cpsdvsr] to set the clock pre-divider.

Step 8 Configure SPI_CR0 to set the serial clock rate, frame formats (SPI, TI, and MW), and frame duration.

Step 9 Write 1 to SPI_CR1[see] to enable the SPI.

Step 10 Query the status register SPI_SR to check the status of the receive FIFO. If the receive FIFO is not empty, read four data segments from the SPI_DR register and receive a half word.

Step 11 Query the status register SPI_SR until the status is not busy.

Step 12 Write 0 to SPI_CR1[see] to disable the SPI.

----End

Data Transfer in DMA Mode

The initialization process is as follows:

Step 1 Write 0 to SPI_CR1[sse] to disable the SPI.

Step 2 Write corresponding values to SPI_CR0 to set the parameters such as the frame format and transfer data bit width.

Step 3 Configure SPI_CPSR to set the required clock divider.

Step 4 Configure the SPI_INTMASK register to disable the generation of the corresponding interrupt signal.

Step 5 Configure SPI_DMACR to enable the DMA function of the SPI.

----End

To transmit data, do as follows:

Step 1 Obtain a DMAC channel.

Step 2 Configure the parameters of the configuration register and control register related to the DMAC channel.

Step 3 Start the DMAC to respond to the DMA request sent by the transmit FIFO of the SPI so as to enable the data transfer.

Step 4 Write 1 to SPI_CR1[sse] to enable the SPI. Then, the initialization configuration is complete.

Step 5 If the number of data segments to be transferred by the DMAC is odd, the DMAC divides the number by the burst length set in Step 2 to calculate the number of burst requests to be responded. Although single requests are generated when the SPI sends all DMA burst requests, the DMAC responds to the burst requests first. In addition, the DMAC responds to the single requests to finish the data transfer only when the last burst is not a complete burst.

Step 6 Check whether the data transfer is complete through the DMA interrupt. If the data transfer is complete, disable the DMA transmit function of the SPI.

----End

To receive data, do as follows:

Step 1 Obtain a DMAC channel.

Step 2 Configure the parameters of the configuration register and control register related to the DMAC channel.

Step 3 Start the DMAC to respond to the DMA request sent by the transmit FIFO of the SPI so as to enable the data transfer.

Step 4 Write 1 to SPI_CR1[sse] to enable the SPI. Then, the initialization configuration is complete.

Step 5 If the number of data segments to be transferred by the DMAC is odd, the DMAC divides the number by the burst length set in Step 2 to calculate the number of burst requests to be responded. Although single requests are generated when the SPI sends all DMA burst

requests, the DMAC responds to the burst requests first. Additionally, the DMAC responds to the single requests to finish the data transfer only when the last burst is not a complete burst.

Step 6 Check whether the data transfer is complete through the DMA interrupt. If the data transfer is complete, disable the DMA transmit function of the SPI.

----End

11.3.6 Register Summary

Table 11-14 lists the SPI registers.

Table 11-14 Summary of the SPI registers (base address: 0x200E_0000)

Offset Address	Register	Description	Page
0x000	SPI_CR0	Control register 0	11-65
0x004	SPI_CR1	Control register 1	11-67
0x008	SPI_DR	Receive/transmit FIFO data register	11-68
0x00C	SPI_SR	FIFO status register	11-68
0x010	SPI_CPSR	Clock pre-divider register	11-69
0x014	SPI_INTMASK	Interrupt mask/clear register	11-69
0x018	SPI_RINTSTATUS	Raw interrupt status register	11-70
0x01C	SPI_MINTSTATUS	Masked interrupt status register	11-71
0x020	SPI_INTCLR	Interrupt clear register	11-72
0x024	SPI_DMACR	DMA control register	11-72

11.3.7 Register Description

SPI_CR0

SPI_CR0 is control register 0. It is used to control various functions of the SPI.

Offset Address	Register Name	Total Reset Value
0x000	SPI_CR0	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	scr								sph	spo	frf		dss			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:8]	RW	scr	Serial clock rate.

			The system clock reference (SCR) value is used to calculate the SPI transmit or receive bit rate. The bit rate is calculated by the following formula: $\frac{F_{SSPCLK}}{CPSDVSR \times (1+SCR)}$ If SPICPSR[CPSDVSR] is an even number ranging from 2 to 254, then the value of SCR ranges from 0 to 255.
[7]	RW	sph	SPICK phase control (applicable to the Motorola SPI frame format only). 0: Data is captured on the first clock edge. 1: Data is captured on the second clock edge.
[6]	RW	spo	SPICK level (applicable to the Motorola SPI frame format only). 0: The level of the SPICK pin is stably low. 1: When there is no data to transmit, the level of the SPICK pin is stably high.
[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.
[3:0]	RW	dss	Data size select. 0000–0010: reserved 0011: 4 bits 0100: 5 bits 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

SPI_CR1

SPI_CR1 is control register 1. It is used to control various functions of the SPI.

Offset Address	Register Name	Total Reset Value
0x004	SPI_CR1	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												sod	ms	sse	lbm
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:4]	RO	reserved	Reserved.
[3]	RW	sod	Slave mode output disable. The bit is used only in slave mode (ms = 1). In a system with multiple slave devices, the SPI master device broadcasts the information to all the slave devices to ensure that only one slave device can drive the data to its serial output line. In this case, the slave output device is connected to SPI_SPIDI. In such a system, if the slave device does not drive SPI_SPIDO, the sod bit must be set. 0: The SPI can drive the SPITXD output in slave mode. 1: The SPI cannot drive the SPITXD output in slave mode.
[2]	RW	ms	Master/slave mode select. The bit can be set only when the SPI is not enabled. 0: The device is configured as a master device (default). 1: The device is configured as a slave device.
[1]	RW	sse	Synchronous serial interface enable. 0: The interface is disabled 1: The interface is enabled.
[0]	RW	lbm	Loopback mode. 0: The normal serial interface operation is enabled. 1: The transmit serial shift register is internally connected to the receive serial shift register.

SPI_DR

SPI_DR is the receive/transmit FIFO data register. When SPI_DR is read, it serves as the exit of the receive FIFO. When the SPI_DR is written, it serves as the entrance of the transmit FIFO. In the Microwire frame format, the data in the transmit FIFO has a fixed bit width of 8 bits. In data receive, the bit width is not restricted. If the SPI_CR1[sse] is set to 0, the receive and transmit FIFOs are not cleared. Therefore, the data can be moved to the transmit FIFO before the SPI starts.

Offset Address	Register Name	Total Reset Value
0x008	SPI_DR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	data															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:0]	RW	data	In the case of read, the register acts as the receive FIFO. In the case of write, the register acts as the transmit FIFO.

SPI_SR

SPI_SR is the FIFO status register. This register is read-only.

Offset Address	Register Name	Total Reset Value
0x00C	SPI_SR	0x0003

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved											bsy	rff	rne	tnf	tfe
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	Access	Name	Description
[15:5]	RO	reserved	Reserved.
[4]	RO	bsy	SPI busy flag. 0: The SPI is idle. 1: The SPI is busy.
[3]	RO	rff	Receive FIFO full state. 0: The receive FIFO is not full. 1: The receive FIFO is full.
[2]	RO	rne	Receive FIFO empty state. 0: The receive FIFO is empty. 1: The receive FIFO is not empty.
[1]	RO	tnf	Transmit FIFO full state.

			0: The transmit FIFO is full. 1: The transmit FIFO is not full.
[0]	RO	tfe	Transmit FIFO empty state. 0: The transmit FIFO is not empty. 1: The transmit FIFO is empty.

SPI_CPSR

The SPI_CPSR is the clock pre-divider register. It specifies a clock pre-divider that divides the frequency of the input SPI working clock to generate SPICK. The pre-divider must be an even number ranging from 2 to 254. The LSB must be zero. If an odd number is written to SPI_CPSR, the returned LSB must be zero when the register is read.

Offset Address	Register Name	Total Reset Value
0x010	SPI_CPSR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								cpsdvsr							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:8]	RO	reserved	Reserved.
[7:0]	RW	cpsdvsr	Clock pre-divider. The value must be an even number ranging from 2 to 254. Either the pre-divider or SPI_CR0[scr] can divide the frequency of the input clock SPICLK. The LSB of CPSDVSR is read as 0.

SPI_INTMASK

SPI_INTMASK is the interrupt mask/clear register. Writing to the corresponding bit of the register can mask or clear an interrupt.

Offset Address	Register Name	Total Reset Value
0x014	SPI_ INTMASK	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												txim	rxim	rtim	rorim
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description

[15:4]	RO	reserved	Reserved.
[3]	RW	txim	Interrupt that is generated when the transmit FIFO is more than half empty. 0: The interrupt is masked. 1: The interrupt is not masked.
[2]	RW	rxim	Interrupt that is generated when the receive FIFO is more than half empty. 0: The interrupt is masked. 1: The interrupt is not masked.
[1]	RW	rtim	Timeout interrupt that is generated when the receive FIFO is not empty and the receive FIFO is not read before the timeout period ends. 0: The interrupt is masked. 1: The interrupt is not masked.
[0]	RW	rorim	Overflow interrupt that is generated when data is written to the full receive FIFO. 0: The interrupt is masked. 1: The interrupt is not masked.

SPI_RINTSTATUS

SPI_RINTSTATUS is the raw interrupt status register. When it is read, the status of the raw interrupts is obtained. Writing this register has no effect.

Offset Address	Register Name	Total Reset Value
0x018	SPI_RINTSTATUS	0x0008

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												txris	rxris	ptris	rorris
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	Access	Name	Description
[15:4]	RO	reserved	Reserved.
[3]	RO	txris	Raw interrupt status of the SPITXINTR interrupt. 0: The interrupt is invalid. 1: The interrupt is valid.
[2]	RO	rxris	Raw interrupt status of the SPIRXINTR interrupt.

			0: The interrupt is invalid. 1: The interrupt is valid.
[1]	RO	rtris	Raw interrupt status of the SPIRTXINTR interrupt. 0: The interrupt is invalid. 1: The interrupt is valid.
[0]	RO	rorris	Raw interrupt status of the SPIRORXINTR interrupt. 0: The interrupt is invalid. 1: The interrupt is valid.

SPI_MINTSTATUS

SPI_MINTSTATUS is the masked interrupt status register. This register is read-only.

Offset Address	Register Name	Total Reset Value
0x01C	SPI_MINTSTATUS	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												txmis	rxmis	rtmis	rormis
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:4]	RO	reserved	Reserved.
[3]	RO	txmis	SPITXINTR interrupt that indicates the transmit FIFO masked interrupt status. 0: The interrupt is invalid. 1: The interrupt is valid.
[2]	RO	rxmis	SPIRXINTR interrupt that indicates the receive FIFO masked interrupt status. 0: The interrupt is invalid. 1: The interrupt is valid.
[1]	RO	rtmis	SPIRTINTR interrupt that indicates the receive timeout masked interrupt status. 0: The interrupt is invalid. 1: The interrupt is valid.
[0]	RO	rormis	SPIRORINTR interrupt that indicates the receive overflow masked interrupt status. 0: The interrupt is invalid.

			1: The interrupt is valid.

SPI_INTCLR

SPI_INTCLR is the interrupt clear register. When 1 is written to the register, the corresponding interrupt is cleared. Writing 0 to the register has no effect.

Offset Address	Register Name	Total Reset Value
0x020	SPI_INTCLR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved														rtic	roric
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:2]	RO	reserved	Reserved.
[1]	WO	rtic	SPIRTINTR interrupt clear. 0: The interrupt is not cleared. 1: The interrupt is cleared.
[0]	WO	roric	SPIRORINTR interrupt clear. 0: The interrupt is not cleared. 1: The interrupt is cleared.

SPI_DMACR

SPI_DMACR is the DMA control register. It is used to enable the DMA request function of the SPI.

Offset Address	Register Name	Total Reset Value
0x024	SPI_DMACR	0x0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved														txdmae	rxdmae
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[15:2]	RO	reserved	Reserved.

[1]	RW	txdmae	DMA operation enable of the transmit FIFO. 0: The DMA operation is disabled. 1: The DMA operation is enabled.
[0]	RW	rxdmae	DMA operation enable of the receive FIFO. 0: The DMA operation is disabled. 1: The DMA operation is enabled.

11.4 IR

11.4.1 Overview

The infrared remoter (IR) module receives the infrared data through the infrared interface. It supports the functions of data decoding in four formats including NEC with simple repeat code, NEC with full repeat code, SONY code, and TC9012 code. It also supports the functions of receive data error detection and IR wake-up.

11.4.2 Features

The IR module has the following features:

- Can be disabled through the configuration by software.
- Supports the receive data overflow interrupt, receive data frame format error interrupt, receive data frame interrupt, key release interrupt, and the interrupt formed by the combination of any the preceding four interrupts.
- Supports the query of the raw interrupt state and the query of the masked interrupt state.
- Supports interrupt clear and mask (write to clear).
- Supports IR wake-up.
- Supports the optional reference clock frequency ranging from 1 MHz to 128 MHz. Through the clock divider programmed by software, the frequency of the working clock can be preset to 1 MHz.

11.4.3 Signal Description

Table 11-15 lists IR interface signals.

Table 11-15 IR interface signals

Signal Name	Direction	Description	Corresponding Pin
IR_RCV	I	Serial IR data received from the pins.	EBIRDYN

a: When the IRRCV pin is used for receiving IR signals, it must be connected to a pull-up resistor.

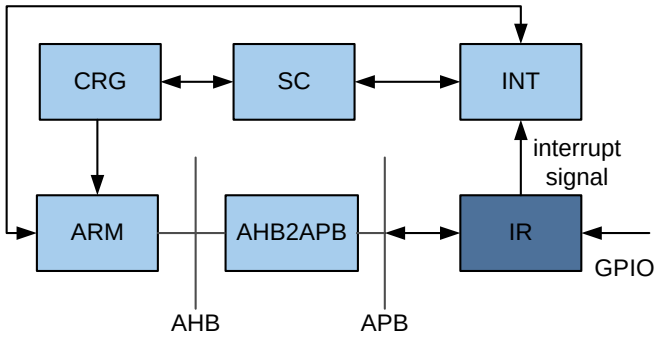
11.4.4 Function Description

The IR module receives the infrared signals transmitted from the infrared remote control, decodes the signals, and then transmits the decoded signals to the ARM system. The ARM system performs the corresponding operations according to the received codes and

implements the expected function. The IR module connects to the advanced peripheral bus (APB) of the ARM subsystem. When the chip is in the low power state (the CPU stops working), the IR module generates an interrupt signal after the receipt of a complete frame and then sends the signal to the interrupt system (INT) module. The INT module sends a control signal to control the system controller and thus to control the clock reset generation (CRG) to wake the ARM subsystem. In this manner, the IR wake function is implemented.

Figure 11-23 shows the functional block diagram of the IR module.

Figure 11-23 Functional block diagram of the IR module



The analysis of signals transmitted from various infrared remote controls shows that the lead codes in the infrared commands vary with remote controls. In addition, the subsequent control commands and even the bits of the command codes are also different. The reason is that the design of these infrared remote controls does not comply with a unified infrared remote control standard. The basic encoding principles, however, are the same. That is, use the pulses with different periods and duty ratios to represent 0 and 1 respectively. The duty ratios and pulse cycles may vary with remote controls. According to these differences, the code formats of the infrared data are classified into NEC with simple repeat code, NEC with full repeat code, TC9012 code, and SONY code.

Table 11-16 to Table 11-18 describe the statistics on the code formats of the infrared receive data.

Table 11-16 Statistics on the code formats of the infrared receive data (NEC with simple repeat code)

Data Format		NEC with Simple Repeat Code			
		uPD6121G	D6121/BU5777 /D1913	LC7461M-C13	AEHA
Lead code (10 µs)	LEAD_S	900	900	900	337.6
	LEAD_E	450	450	450	168.8
bit0 (10 µs)	B0_L	56	56	56	42.2
	B0_H	56	56	56	42.2
bit1 (10 µs)	B1_L	56	56	56	42.2
	B1_H	169	169	169	126.6

Data Format		NEC with Simple Repeat Code			
		uPD6121G	D6121/BU5777 /D1913	LC7461M-C13	AEHA
Simple repeat code (10 µs)	SLEAD_S	900	900	900	337.6
	SLEAD_E	225	225	225	337.6
Burst (10µs)		55	55	55	42.2
Frame duration (10 µs)		10800	10800	10800	8777.6–12828.8
Valid data bit		32	32	42	48

Table 11-17 Statistics on the code formats of the infrared receive data (NEC with full repeat code)

Data Format		NEC with Full Repeat Code						
		uPD6121G	LC7461M-C13	MN6024-C5D6	MN6014-C6D6	MATNEW	MN6030	PANASONIC
Lead code (10 µs)	LEAD_S	900	900	337.6	349.2	348.8	349	352
	LEAD_E	450	450	337.6	349.2	374.4	349	352
bit0 (10 µs)	B0_L	56	56	84.4	87.3	43.6	87.3	88
	B0_H	56	56	84.4	87.3	43.6	87.3	88
bit1 (10 µs)	B1_L	56	56	84.4	87.3	43.6	87.3	88
	B1_H	169	169	253.2	174.6	130.8	261.9	264
simple repeat code (10 µs)	SLEAD_S	None	None	None	None	None	None	None
	SLEAD_E							
Burst (10 µs)		55	55	84.4	87.3	43.6	87.3	88
Frame duration (10µs)		10800	10800	10130	10470	12413.6–16594.4	10500	10400
Valid data bit		32	42	22	24	48	22	22

Table 11-18 Statistics on the code formats of the infrared receive data (TC9012 and SONY)

Data Format		TC9012	SONY			
		TC9012F/9243	SONY-D7C5	SONY-D7C6	SONY-D7C8	SONY-D7C13
Lead code (10 μs)	LEAD_S	450	240	240	240	240
	LEAD_E	450	60	60	60	60
bit0 (10 μs)	B0_L	56	60	60	60	60
	B0_H	56	60	60	60	60
bit1 (10 μs)	B1_L	56	120	120	120	120
	B1_H	169	60	60	60	60
Simple repeat code (10 μs)	SLEAD_S	None	None	None	None	None
	SLEAD_E					
Burst (10 μs)		56	None	None	None	None
Frame duration (10 μs)		10800	4500	4500	4500	4500
Valid data bit		32	12	13	15	20

NEC with Simple Repeat Code

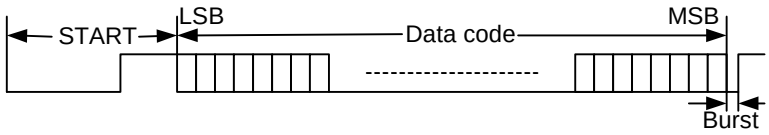
Frame Format

The NEC with simple repeat code consists of three parts:

- START (lead code): It consists of a start code (low level) and an end code (high level).
- Data code: The valid bit and the meaning of a certain bit depend on the specific code format. The data code is received according to the least significant bit (LSB) first sequence.
- Burst: It is used to receive the last data bit.

Figure 11-24 shows the frame format for transmitting a single NEC with simple repeat code.

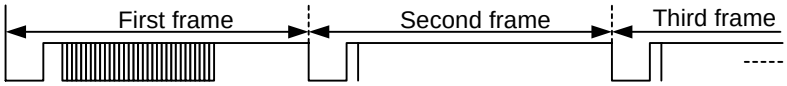
Figure 11-24 Frame format for transmitting a single NEC with simple repeat code



When a complete data frame is received after the key is held down for more than one frame duration, the following received data frame consists of a simple lead code and burst signal only. The lead code also consists of a start code (low level) and an end code (high level).

Figure 11-25 shows the frame format for transmitting continuous NEC with simple repeat codes by holding the key down.

Figure 11-25 Frame format for transmitting continuous NEC with simple repeat codes by holding the key down



Code Format

Figure 11-26 shows the definitions of bit0 and bit1 of the NEC with simple repeat code.

Figure 11-26 Definitions of bit0 and bit1 in the NEC with simple repeat code

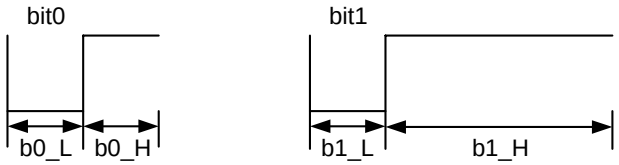


Figure 11-27 shows the code format for transmitting a single NEC with simple repeat code.
Figure 11-28 shows the code format for transmitting continuous NEC with simple repeat codes by holding the key down.

Figure 11-27 Code format for transmitting a single NEC with simple repeat code

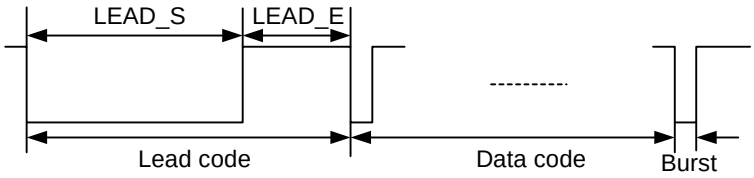
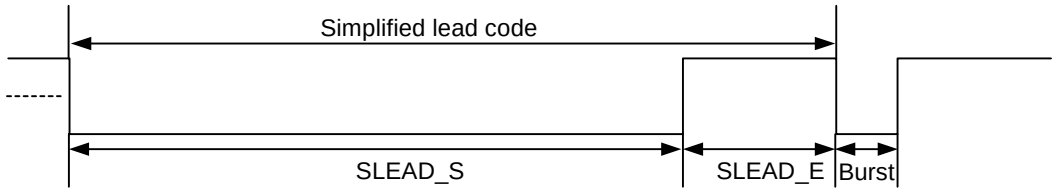


Figure 11-28 Code format for transmitting continuous NEC with simple repeat codes by holding the key down



Note 1: The pulse width of the high and low levels and the frame duration depend on specific code formats. See Table 11-16 to Table 11-18.
Note 2: The frame duration must be equal to or less than 160 ms. Otherwise, the simple lead code cannot be identified.

NEC with Full Repeat Code

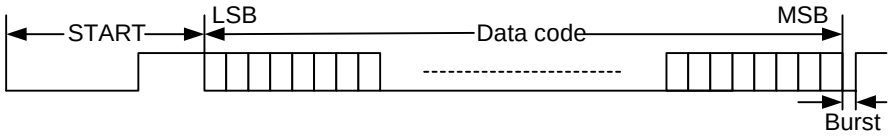
Frame Format

The NEC with full repeat code consists of three parts:

- START (lead code): It consists of a start code (low level) and an end code (high level).
- Data code: The valid bit and the meaning of a certain bit depend on the specific code format. The data code is received according to the LSB first sequence.
- Burst: It is used to receive the last data bit.

Figure 11-29 shows the frame format for transmitting a single NEC with full repeat code.

Figure 11-29 Frame format for transmitting a single NEC with full repeat code



When a complete data frame (first frame) is received after the key is held down for more than one frame duration, the following received data frames are all complete data frames. That is, the first frame is transmitted repeatedly based on the frame duration. Figure 11-30 shows the frame format for transmitting continuous NEC with full repeat codes by holding the key down.

Figure 11-30 Frame format for transmitting continuous NEC with full repeat codes by holding the key down

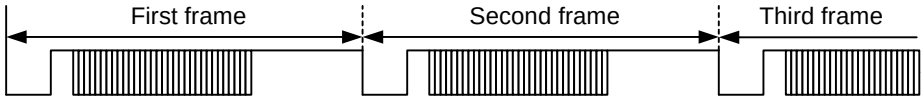


Figure 11-28 and Figure 11-30 show that the only difference between the NEC with simple repeat code and the NEC with full repeat code is the format of the repeat frame. For the NEC with simple repeat code, the transmitted repeat frame includes a simple lead code; for the NEC with full repeat code, the transmitted repeat frame is a complete frame, that is, the first frame and the repeat frames are the same.

Code Format

Figure 11-31 shows the definitions of bit0 and bit1 of the NEC with full repeat code.

Figure 11-31 Definitions of bit0 and bit1 in the NEC with full repeat code

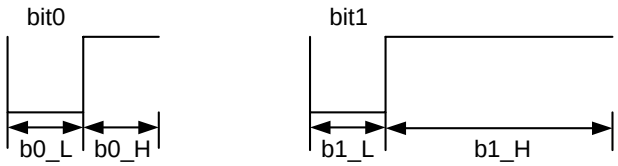
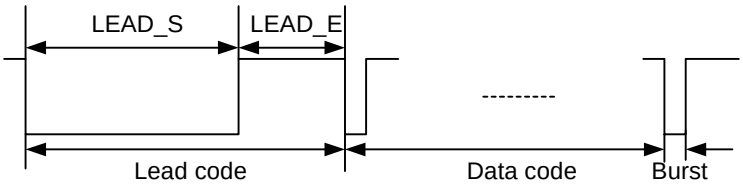


Figure 11-32 shows the code format for transmitting a single NEC with full repeat code.

Figure 11-32 Code format for transmitting a single NEC with full repeat code



Note: The pulse width of the high and low levels and the frame duration depend on specific code formats. See Table 11-16 to Table 11-18.

TC9012 Code

Frame Format

CAUTION

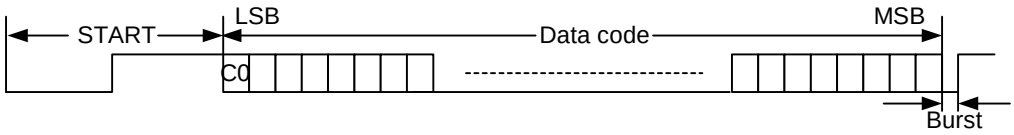
According to the features of the TC9012 code, the first bit of all key codes must be all-1s or all-0s. Otherwise, the unnecessary frames are generated due to continuous key press.

The TC9012 code consists of three parts:

- START (lead code): It consists of a start code (low level) and an end code (high level).
- Data code: The valid bit and the meaning of a certain bit depend on the specific code format. The data code is received according to the LSB first sequence.
- Burst: It is used to receive the last data bit.

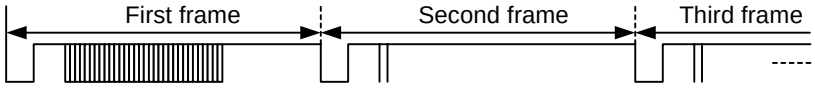
Figure 11-33 shows the frame format for transmitting a single TC9012 code.

Figure 11-33 Frame format for transmitting a single TC9012 code



When a complete data frame is received after the key is held down for more than one frame duration, the following received data frame consists of a simple lead code, data bit, and burst signal. The lead code consists of a start code (low level) and an end code (high level). The data bit is the one's complement of the first data bit (C0) received in the previous frame. Figure 11-34 shows the frame format for transmitting continuous TC9012 codes.

Figure 11-34 Frame format for transmitting continuous TC9012 code by holding the key down



Code Format

Figure 11-35 shows the definitions of bit0 and bit1 of the TC9012 code.

Figure 11-35 Definitions of bit0 and bit1 of the TC9012 code

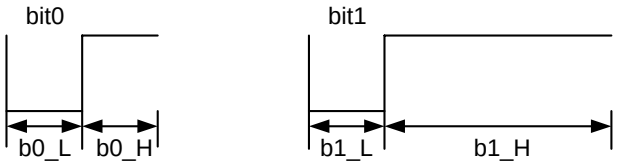


Figure 11-36 shows the code format for transmitting a single TC9012 code.

Figure 11-36 Code format for transmitting a single TC9012 code

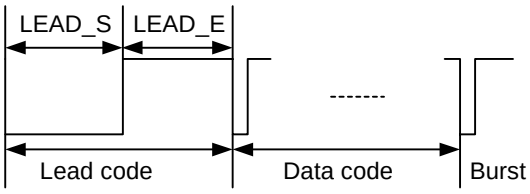


Figure 11-37 shows the code format for transmitting continuous TC9012 codes when C0 is 1.

Figure 11-37 Code format for transmitting continuous TC9012 codes (C0 = 1)

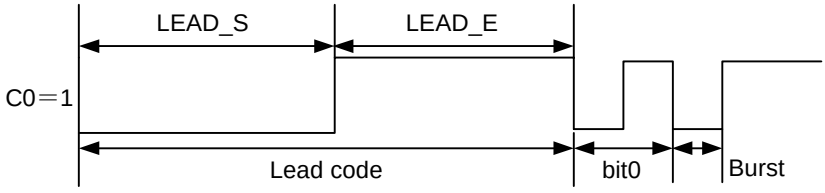
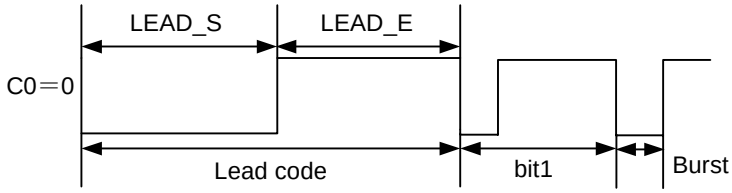


Figure 11-38 shows the code format for transmitting continuous TC9012 codes when C0 is 0.

Figure 11-38 Code format for transmitting continuous TC9012 codes (C0 = 0)



Note: The pulse width of the high and low levels and frame duration depend on specific code patterns. See Table 11-16 to Table 11-18. In addition, the frame duration must be equal to or less than 160 ms. Otherwise, the repeat frame cannot be identified.

SONY Code

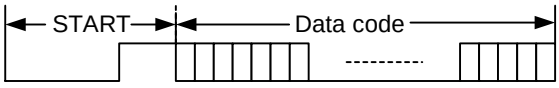
Frame Format

The SONY code consists of two parts:

- START (lead code): It consists of a start code (low level) and an end code (high level).
- Data code: The valid bit and the meaning of a certain bit depend on the specific code format. The data code is received according to the LSB first sequence.

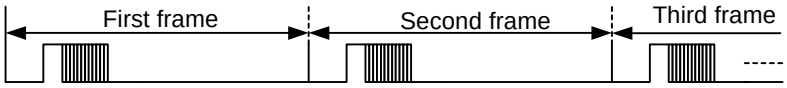
Figure 11-39 shows the frame format for transmitting a single SONY code.

Figure 11-39 Frame format for transmitting a single SONY code



When a complete data frame is received after the key is held down for more than one frame duration, the following received data frame is also a complete data frame. Figure 11-40 shows the frame format for transmitting continuous SONY codes by holding the key down.

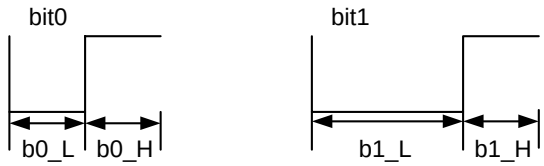
Figure 11-40 Frame format for transmitting continuous SONY codes by holding the key down



Code Format

Figure 11-41 shows the definitions of bit0 and bit1 of the SONY code.

Figure 11-41 Definitions of bit0 and bit1



Note: The pulse width of the high level and low level and frame duration depend on specific code patterns. For details, see Table 11-16 to Table 11-18.

11.4.5 Operating Mode

Pin Multiplexing

The IR pin is multiplexed with the EBI pin. Before using the IR pin, you must configure the IO Config register to enable the IR function of the corresponding pin. For details, see the configuration of reg45.

Clock Gating

When SC_PEREN bit[11] is set to 1, the IR clock pclk is enabled. When SC_PERDIS bit[11] is set to 1, the IR clock pclk is disabled.

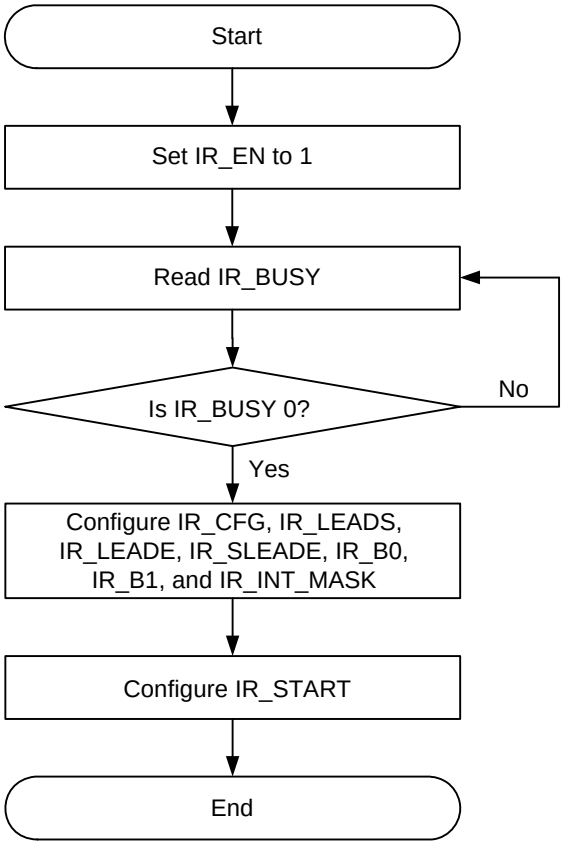
Soft Reset

When SC_PERCTRL8 bit[10] is set to 1, the IR module is soft reset separately. After the reset, each configuration register is reset to its default value. Therefore, these registers must be initialized after the reset.

Instance of Register Configuration

Figure 11-42 shows the process of initializing the IR module.

Figure 11-42 Process of initializing the IR module



To initialize the IR module, do as follows:

Step 1 Start the initialization process after selecting the address space of the IR module.

Step 2 Configure IR_EN[0] to 1 to enable the IR receive module.

Step 3 Read IR_BUSY to check the current configuration state of the IR module.

- If the value of IR_BUSY is 1, it indicates that the IR module is in the busy state. Then keep reading IR_BUSY.

  In this case, do not configure other control registers of the IR module through software. Otherwise, the IR module fails to be configured.
- If the value of IR_BUSY is 0, it indicates that the IR module is in the idle state. Then go to Step 4.

Step 4 Configure IR_CFG, IR_LEADS, IR_LEADE, IR_SLEADE, IR_B0, IR_B1, and IR_INT_MASK.

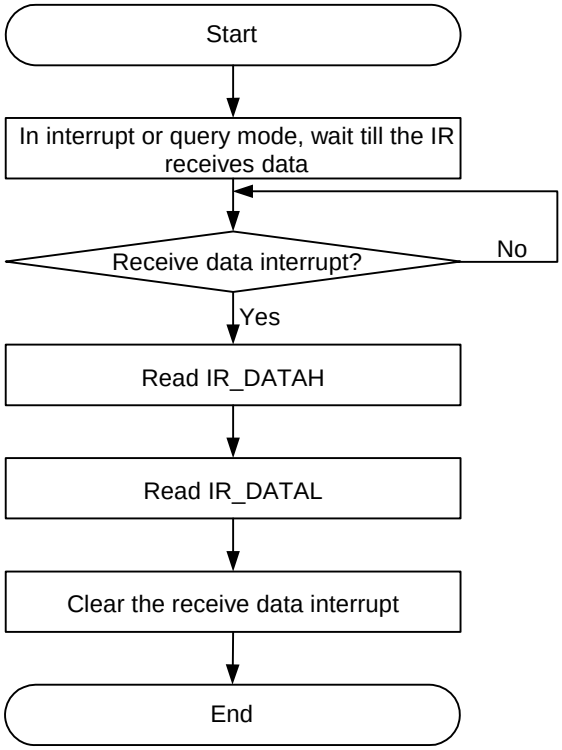
Note that users can update corresponding registers as required. If the registers are not updated, the original values are remained.

Step 5 Configure IR_START.

The IR_START can be configured until all IR control registers are configured, because the IR_START is used to generate the start signal. If the IR_START is configured, the IR module starts to receive the infrared data according to the register value.

----End

Figure 11-43 Process of reading the decoded data



To read the decoded data, do as follows:

Step 1 Select the address space of the IR module.

Step 2 In interrupt or query mode, wait for the receive data frame.

- In interrupt mode, the CPU starts to query the value of IR_INT_STATUS[intms_rcv] when it receives an interrupt request signal from the IR module. If the queried value is 1, it indicates that the IR module receives a data frame. Then, go to Step 3. If the queried value is 0, repeat Step 2 to wait for an interrupt.
- In query mode, software keeps querying the value of IR_INT_STATUS[intrs_rcv] or queries the value at an interval. If the queried value is 1, it indicates that the IR module receives a data frame. Then, go to Step 3. If the queried value is 0, it indicates that the IR module does not receive a data frame yet. Then, repeat Step 2 to continue the query.

Step 3 Read IR_DATAH.

If the data in one frame does not exceed 32 bits, skip this step.

Step 4 Read IR_DATAL.

Step 5 Clear the receive data interrupt.

----End

11.4.6 Register Summary

Table 11-19 lists the IR registers.

Table 11-19 Summary of the IR registers (base address: 0x2007_0000)

Offset Address	Register	Description	Page
0x000	IR_EN	IR receive enable control register	11-85
0x004	IR_CFG	IR configuration register	11-86
0x008	IR_LEADS	Margin configuration register of the start bit in the lead code	11-87
0x00C	IR_LEADE	Margin configuration register of the end bit in the lead code	11-88
0x010	IR_SLEADE	Margin configuration register of the end bit in the simple lead code	11-89
0x014	IR_B0	Margin configuration register of the level for determining bit0	11-91
0x018	IR_B1	Margin configuration register of the level for determining bit1	11-92
0x01C	IR_BUSY	Configuration busy flag register	11-93
0x020	IR_DATAH	Register for storing the 16 most significant bits (MSBs) of the decoded data received by the IR	11-93
0x024	IR_DATAL	Register for storing the 32 LSBs of the decoded data received by the IR	11-94
0x028	IR_INT_MASK	IR interrupt mask register	11-94
0x02C	IR_INT_STATUS	IR interrupt status register	11-95
0x030	IR_INT_CLR	IR interrupt clear register	11-96
0x034	IR_START	IR start configuration register	11-97

11.4.7 Register Description

IR_EN

IR_EN is the IR receive enable control register.

CAUTION

You must set IR_EN[0] to 0b1 through software before configuring other registers. Otherwise, the configuration is invalid. When IR_EN[0] is 0b0, other registers are read-only and the read values are their reset values.

Offset Address	Register Name	Total Reset Value
0x000	IR_EN	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RW	ir_en	IR receive module enable bit. 0: The IR receive module is disabled. 1: The IR receive module is enabled.

IR_CFG

IR_CFG is the IR configuration register.

CAUTION

Before configuring this register, ensure that IR_BUSY[0] is 0b0 and IR_EN[0] is 0b1. Otherwise, the configuration is invalid and the register remains the original value.

The reference clock frequency supported by the IR module ranges from 1 MHz to 128 MHz. The relation between the frequency and the divider ir_freq is as follows:

- When the reference clock frequency is 1 MHz. ir_freq must be set to 0x00.
- If the reference clock frequency is 128 MHz. ir_freq must be set to 0x7F.

When the frequency of the IR reference clock is not an integer ranging from 1 MHz to 128 MHz, the divider can be calculated by the round-off method. For example, if the reference clock is 12.1 MHz, the divider is 0x0B; if the reference clock is 12.8 MHz, the divider is 0x0C.

The relation between the frequency tolerance and the count difference is as follows: If the base frequency is f and the frequency difference is D_f, then the frequency tolerance ratio is D_f/f. If the count difference is represented by Dcnt and the level width is s (in the unit of μs), then the count difference is: $Dcnt = \lceil 0.1 \times s \times ratio \rceil$. Therefore, when the clock has a

frequency tolerance, the valid range of the parameter value offsets. If the frequency increases, the corresponding margin value range is from (min + Dcnt) to (max + Dcnt). The symbols min and max represent the margin values without the offset. If the frequency decreases, the offset range is from (min – Dcnt) to (max – Dcnt). Here, take the margin of the start bit in the lead code as an example. If the base frequency is 100 MHz and the frequency is increased by 0.1 MHz, then ratio is 0.1/100 (0.001). Assume that s is equal to 9000 μs. Then $Dcnt = \lceil 0.1 \times 9000 \times 0.001 \rceil = 1$. Thus, the margin of ir_leads must be changed to a value ranging from 0x033D to 0x3CD.

Offset Address	Register Name	Total Reset Value
0x004	IR_CFG	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:14]	RW	ir_format	Data code type. 00: NEC with simple repeat code 01: TC9012 code 10: NEC with full repeat code 11: SONY code For details about the relations between the codes and code types, see Table 11-16 to Table 11-18.
[13:8]	RW	ir_bits	Data bits in a frame. 0x00–0x2F: Correspond to bit 1 to bit 48 in one frame respectively. 0x30–0x3F: Reserved. If software configures ir_bits to the value ranging from 0x30 to 0x3F, the configuration is invalid and ir_bits remains unchanged.
[7]	RO	reserved	Reserved.
[6:0]	RW	ir_freq	Divider of the working clock. 0x00–0x7F: Correspond to the working clock divider 1–128 respectively.

IR_LEADS

IR_LEADS is the margin configuration register of the start bit in the lead code.

CAUTION

Before configuring this register, ensure that IR_BUSY[0] is 0b0 and IR_EN[0] is 0b1. Otherwise, the configuration is invalid and the register remains the original value.

To accurately determine the start bit of the lead code, the margin must be set to the typical value of the specific code type. For details about the typical values of specified code types, see LEAD_S in Table 11-16 to Table 11-18.

- For a pulse width whose typical value is equal to or greater than 400 (with the precision of 10 μs), the recommended margin is 8% of the typical value. For example, assume that the code type is D6121 and the typical value of LEAD_S is 900, then cnt_leads_min = 900 x 92% = 828 = 0x33C and cnt_leads_max = 900 x 108% = 972 = 0x3CC.
- For a pulse width whose typical value is less than 400 (with the precision of 10 μs), the recommended margin is 20% of the typical value. For example, if the code type is SONY-D7C5 and the typical value of LEAD_S is 240, then cnt_leads_min = 240 x 80% = 192 = 0xC0 and cnt_leads_min = 240 x 120% = 288 = 0x120.

The basic configuration principle is as follows: cnt_leads_max is equal to or greater than cnt_leads_min, and cnt_leads_min is greater than cnt0_b_max and cnt1_b_max.

Offset Address	Register Name	Total Reset Value
0x008	IR_LEADS	0x033C_03CC

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

Bits	Access	Name	Description
[25:16]	RW	cnt_leads_min	Minimum pulse width of the start bit in the lead code. 0x000–0x007: Reserved.
[15:10]	RO	reserved	Reserved.
[9:0]	RW	cnt_leads_max	Maximum pulse width of the start bit of the lead code. 0x000–0x007: Reserved.

IR_LEADE

IR_LEADE is the margin configuration register of the end bit in the lead code.

CAUTION

- Before configuring this register, ensure that IR_BUSY[0] is 0b0 and IR_EN[0] is 0b1. Otherwise, the configuration is invalid and the register remains the original value.
- For the NEC with simple repeat code, the margins of cnt_sleade and cnt_leade cannot be overlapped. Otherwise, when the actual count value is in the overlapped range, the simplified lead code cannot be identified. Therefore, a frame format error occurs.

To accurately determine the end bit of the lead code, the margin must be set to the typical value of the specific code type. Generally, the margin is 8% of the typical value. For details about the typical values of specified code types, see the value of LEAD_E in Table 11-16 to Table 11-18.

- For the pulse width whose typical value is equal to or greater than 400 (with the precision of 10 μs), the recommended margin is 8% of the typical value. For example, if the code type is D6121 and the typical value of LEAD_E is 450, then cnt_leade_min = 450 x 92% = 414 = 0x19E and cnt_leade_max = 450 x 108% = 486 = 0x1E6.
- For a pulse width whose typical value is less than 400 (with the precision of 10 μs), the recommended margin is 20% of the typical value. For example, if the code type is SONY-D7C5 and the typical value of LEAD_E is 60, then cnt_leade_min = 60 x 80% = 48 = 0x030 and cnt_leade_max = 60 x 120% = 72 = 0x048.

The basic configuration principle is as follows: cnt_leade_max is equal to or greater than cnt_leade_min.

Offset Address	Register Name	Total Reset Value
0x00C	IR_LEADE	0x019E_01E6

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

Bits	Access	Name	Description
[31:25]	RO	reserved	Reserved.
[24:16]	RW	cnt_leade_min	Minimum pulse width of the end bit in the lead code. 0x000–0x007: Reserved.
[15:9]	RO	reserved	Reserved.
[8:0]	RW	cnt_leade_max	Maximum pulse width of the end bit in the lead code. 0x000–0x007: Reserved.

IR_SLEADE

IR_SLEADE is the margin configuration register of the end bit in the simplified lead code.

CAUTION

- Before configuring this register, ensure that IR_BUSY[0] is 0b0 and IR_EN[0] is 0b1. Otherwise, the configuration is invalid and the register remains the original value.
- For the NEC with simple repeat code, the margins of cnt_sleade and cnt_leade cannot be overlapped. Otherwise, when the actual count value is in the overlapped range, the simplified lead code cannot be identified. Therefore, a frame format error occurs.
- The register must be configured only when the data format is NEC with simple repeat code.

To accurately determine the end bit of the simplified lead code, the margin must be set to the typical value of the specific code type. For details about the typical values of specified code types, see SLEAD_E in Table 11-16 to Table 11-18

- For a pulse width whose typical value is equal to or greater than 225 (with the precision of 10 μs), the recommended margin is 8% of the typical value. For example, if the code type is D6121 and the typical value of SLEAD_E is 225, then cnt_sleade_min = 225 x 92% = 207 = 0xCF and cnt_sleade_max = 225 x 108% = 243 = 0xF3.
- For a pulse width whose typical value is less than 225 (with the precision of 10 μs), the recommended margin is 20% of the typical value. For example, if the typical value of SLEAD_E of a certain code type is 60, then cnt_sleade_min = 60 x 80% = 48 = 0x30 and cnt_sleade_max = 60 x 120% = 72 = 0x48.

The basic configuration principle is as follows: cnt_sleade_max is equal to or greater than cnt_sleade_min.

Offset Address	Register Name	Total Reset Value
0x010	IR_SLEADE	0x00CF_00F3

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

Bits	Access	Name	Description
[31:25]	RO	reserved	Reserved.
[24:16]	RW	cnt_sleade_min	Minimum pulse width of the end bit in the simplified lead code. 0x000–0x007: Reserved.
[15:9]	RO	reserved	Reserved.
[8:0]	RW	cnt_sleade_max	Maximum pulse width of the start bit of the simplified lead code. 0x000–0x007: Reserved.

IR_B0

IR_B0 is the margin configuration register of the level for determining bit0.

CAUTION

- Before configuring this register, ensure that IR_BUSY[0] is 0b0 and IR_EN[0] is 0b1. Otherwise, the configuration is invalid and the register remains the original value.
- For the described four code types, the margins of the levels for determining bit0 and bit1 cannot be overlapped. Otherwise, the actual count value is regarded as bit0 when it is in the overlapped range.

To accurately determine bit0, the margin must be set to the typical value of the specific code type. The margin is about 20% of the typical value.

- For details about the typical values of the NEC with simple repeat code, NEC with simple repeat code, and TC9012 code, see the value of B0_H in Table 11-16 to Table 11-18. For example, if the code type is D6121 and the typical value of the B0_H is 56 (with the precision of 10 μs), then cnt0_b _min = 56 x 80% = 45 = 0x2D and cnt0_b_max = 56 x 120% = 67 = 0x43.
- For details about the typical value of the SONY code, see the value of B0_L in Table 11-16 to Table 11-18. For example, if the code type is SONY-D7C5 and the typical value of B0_H is 60 (with the precision of 10 μs), then cnt0_b_min = 60 x 80% = 48 = 0x30 and cnt0_b_max = 60 x 120% = 72 = 0x48.

The basic configuration principle is as follows: cnt0_b_max is equal to or greater than cnt0_b_min.

Offset Address	Register Name	Total Reset Value
0x014	IR_B0	0x002D_0043

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								cnt0_b_min								reserved								cnt0_b_max							
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1

Bits	Access	Name	Description
[31:23]	RO	reserved	Reserved.
[22:16]	RW	cnt0_b_min	Minimum pulse width of the level for determining bit0. 0x00–0x07: Reserved.
[15:7]	RO	reserved	Reserved.
[6:0]	RW	cnt0_b_max	Maximum pulse width of the level for determining bit0. 0x00–0x07: Reserved.

IR_B1

IR_B1 is the margin configuration register of the level for determining bit1.

CAUTION

- Before configuring this register, ensure that IR_BUSY[0] is 0b0 and IR_EN[0] is 0b1. Otherwise, the configuration is invalid and the register remains the original value.
- For the described four code types, the margins of the levels for determining bit0 and bit1 cannot be overlapped. Otherwise, the actual count value is regarded as bit0 when it is in the overlapped range.

To accurately determine bit1, the margin must be set to the typical value of the specific code type. The margin is about 20% of the typical value.

- For details about the typical values of the NEC with simple repeat code, NEC with simple repeat code, and TC9012 code, see the values of B1_H in Table 11-16 to Table 11-18. For example, if the code type is D6121 and the typical value of B1_H is 169 (with the precision of 10 μs), then cnt1_b _min = 169 x 80% = 135 = 0x87 and cnt1_b_max = 169 x 120% = 203 = 0xCB.
- For details about the typical value of the SONY code, see the values of B1_L in Table 11-16 to Table 11-18. For example, if the code type is SONY-D7C5 and the typical value of B1_L is 120 (with the precision of 10 μs), then cnt1_b_min = 120 x 80% = 96 = 0x60 and cnt1_b_max = 120 x 120% = 144 = 0x90.

The basic configuration principle is as follows: cnt1_b_max is equal to or greater than cnt1_b_min.

Offset Address	Register Name	Total Reset Value
0x018	IR_B1	0x0087_00CB

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								cnt1_b_min								reserved								cnt1_b_max							
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1

Bits	Access	Name	Description
[31:25]	RO	reserved	Reserved.
[24:16]	RW	cnt1_b_min	Minimum pulse width of the level for determining bit1. 0x000–0x007: Reserved.
[15:9]	RO	reserved	Reserved.
[8:0]	RW	cnt1_b_max	Maximum pulse width of the level for determining bit1. 0x000–0x007: Reserved.

IR_BUSY

IR_BUSY is the configuration busy flag register.

Offset Address	Register Name	Total Reset Value
0x01C	IR_BUSY	0x0000_0000

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

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	RO	ir_busy	Busy state flag. 0: Indicates the idle state. In this state, software can configure data. 1: Indicates the busy state. In this state, software cannot configure data.

IR_DATAH

IR_DATAH is the register for storing the 16 MSBs of the decoded data received by the IR.

The IR_DATAH register stores the 16 MSBs of the decoded data received by the IR, whereas IR_DATAL stores the 32 LSBs of the decoded data received by the IR. The valid data bits depend on those contained in a frame of a specific code. For details, see the valid data bits in Table 11-16 to Table 11-18.

Principle of data storage: The data must be stored in IR_DATAH first and then IR_DATAL from MSB to LSB. When IR_DATAL is full, the rest data is stored in IR_DATAH. The unused MSBs are reserved. When software reads the data from the registers, it reads the IR_DATAH first and then IR_DATAL.

Hardware receives all data bits without checking the definition of each data bit. Software is responsible for processing the data bits.

Offset Address	Register Name	Total Reset Value
0x020	IR_DATAH	0x0000_0000

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

Bits	Access	Name	Description
[31:16]	RO	reserved	Reserved.
[15:0]	RO	ir_datah	16 MSBs of the decoded data received by the IR.

IR_DATAL

IR_DATAL is the register for storing the 32 LSBs of the decoded data received by the IR.

Offset Address	Register Name	Total Reset Value
0x024	IR_DATAL	0x0000_0000

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

Bits	Access	Name	Description
[31:0]	RO	ir_datal	32 LSBs of the decoded data received by the IR.

IR_INT_MASK

IR_INT_MASK is the IR interrupt mask register.

CAUTION

Before configuring this register, ensure that IR_EN[0] is 0b1. Otherwise, the configuration is invalid and the register remains the original value. If all interrupts are masked, the IR wake-up function is unavailable.

The definitions of interrupts involved in the register are as follows:

- Receive data overflow interrupt

  If the CPU does not take the current frame while the next frame is already received, the next frame overwrites the current frame and a raw receive data overflow error interrupt request is reported.
- Receive data frame format error interrupt

  If the received data frame is not complete or the data pulse width does not meet the margin requirements, a raw receive frame format error interrupt request is reported.
- Receive data frame interrupt

  When a complete frame data is received, a raw receive data frame interrupt request is reported.
- Key release detection interrupt

  For the NEC with simple repeat code and TC9012 code, if the start synchronous code is not detected again within 160 ms after the previously detected start synchronous code, or the detected valid data frame is not a simplified lead code, a raw key release detection interrupt is reported. Neither the NEC with full repeat code nor the SONY code supports the key release interrupt.

The hardware does not identify the interrupt priority. An interrupt can be generated only if any masked interrupt source is valid.

Offset Address	Register Name	Total Reset Value
0x028	IR_INT_MASK	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																												intm_release	intm_overflow	intm_framerr	intm_rcv
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:4]	RO	reserved	Reserved.
[3]	RW	intm_release	Key release interrupt request enable. 0: The interrupt request is enabled. 1: The interrupt request is disabled.
[2]	RW	intm_overflow	Receive data overflow error interrupt request enable. 0: The interrupt request is enabled. 1: The interrupt request is disabled.
[1]	RW	intm_framerr	Receive frame format error interrupt request enable. 0: The interrupt request is enabled. 1: The interrupt request is disabled.
[0]	RW	intm_rcv	Receive data frame interrupt request enable. 0: The interrupt request is enabled. 1: The interrupt request is disabled.

IR_INT_STATUS

IR_INT_STATUS is the IR interrupt status register.

Offset Address	Register Name	Total Reset Value
0x02C	IR_INT_STATUS	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												intms_release	intms_overflow	intms_framerr	intms_rcv	reserved												intrs_release	intrs_overflow	intrs_framerr	intrs_rcv

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:20]	RO	reserved	Reserved.
[19]	RO	intms_release	Masked key release interrupt status. 0: An interrupt is not generated. 1: An interrupt is generated.
[18]	RO	intms_overflow	Masked receive data overflow error interrupt status. 0: An interrupt is not generated. 1: An interrupt is generated.
[17]	RO	intms_framerr	Masked receive frame format error interrupt status. 0: An interrupt is not generated. 1: An interrupt is generated.
[16]	RO	intms_rcv	Masked receive data frame interrupt status. 0: An interrupt is not generated. 1: An interrupt is generated.
[15:4]	RO	reserved	Reserved.
[3]	RO	intrs_release	Raw key release interrupt status. 0: An interrupt is not generated. 1: An interrupt is generated.
[2]	RO	intrs_overflow	Raw receive data overflow error interrupt status. 0: An interrupt is not generated. 1: An interrupt is generated.
[1]	RO	intrs_framerr	Raw receive frame format error interrupt status. 0: An interrupt is not generated. 1: An interrupt is generated.
[0]	RO	intrs_rcv	Raw receive data frame interrupt state. 0: An interrupt is not generated. 1: An interrupt is generated.

IR_INT_CLR

IR_INT_CLR is the IR interrupt clear register.

Offset Address	Register Name	Total Reset Value
0x030	IR_INT_CLR	0x0000_0000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																												intc_release	intc_overflow	intc_framerr	intc_rcv
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:4]	RO	reserved	Reserved.
[3]	WO	intc_release	Key release interrupt request clear. 0: The interrupt request is not cleared. 1: The interrupt request is cleared.
[2]	WO	intc_overflow	Receive data overflow error interrupt request clear. 0: The interrupt request is not cleared. 1: The interrupt request is cleared.
[1]	WO	intc_framerr	Receive frame format error interrupt request clear. 0: The interrupt request is not cleared. 1: The interrupt request is cleared.
[0]	WO	intc_rcv	Receive data frame interrupt request clear. 0: The interrupt request is not cleared. 1: The interrupt request is cleared. If the receive data frame interrupt request is generated and software directly writes 1 to the bit without reading the data in IR_DATAL, then the interrupt request cannot be cleared.

IR_START

IR_START is the IR start configuration register.

After the values of other registers are configured, the IR start configuration register starts when any data is written to the address during the starting of the IR module.

Offset Address	Register Name	Total Reset Value
0x034	IR_START	0x0000_0000

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

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[31:1]	RO	reserved	Reserved.
[0]	WO	ir_start	IR start configuration register.

11.5 GPIO

11.5.1 Overview

The Hi3515 supports six sets of general purpose input/output (GPIO) pins. Each set of GPIO pins provides eight programmable input/output pins. Each pin can be configured as input or output. These pins generate the output signals used for specific applications or capture the input signals used for specific applications. When configured as input, the GPIO can act as the interrupt source. When configured as output, each GPIO can be independently set to 1 or 0. Each GPIO is multiplexed with other pins. For details about pin multiplexing, see section 2.2 "Description of Software Multiplexing Pins." For details about the control information, see section 2.4 "Summary of the IO Config (Pin Multiplexing Control) Registers."

11.5.2 Features

The GPIO module has the following features:

- Each GPIO pin can be configured as input, output, or open drain (OD) output.
  - When the GPIO works as an input pin, it can be used as the interrupt source. That is, each GPIO can generate the interrupt independently.
  - When the GPIO works as an output pin, each GPIO pin can be independently set to 1 or 0.
  - When the output of the GPIO works as an open drain output pin, the board must be connected to a pulled-up resistor. Through the GPIO_DIR register, the board-level wire-AND function is implemented.
- The GPIO interrupts are controlled through seven registers such as GPIO_IS. Through these registers, the interrupt source, interrupt edge (falling edge or rising edge), and interrupt trigger modes (level-sensitive mode or edge-sensitive mode) can be selected. For details about the corresponding interrupt registers of the GPIO, see section 3.4 "Interrupt System."
  - When multiple interrupts occur at the same time, these interrupts are combined as one interrupt to be reported. For details about the GPIO interrupt mapping, see Table 3-16 "Interrupt Mapping."
  - The GPIO_IS, GPIO_IBE, and GPIO_IEV 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. The GPIO_IE controls the final report of the interrupt. In addition, the independent GPIO_IC is provided for clearing the interrupt status.

11.5.3 Signal Description

Table 11-20 lists the GPIO interface signals. For details about the multiplexing configuration information, see "Pin Multiplexing" in section 11.5.5 "Operating Mode."

Table 11-20 GPIO interface signals

Signal	Direction	Description	Pin
GPIO0_0	I/O	GPIO bidirectional data signal, multiplexed with the I^2C data signal	SDA
GPIO0_1	I/O	GPIO bidirectional data signal, multiplexed with the I^2C clock signal	SCL
GPIO0_2	I/O	GPIO bidirectional data signal, multiplexed with the I^2S transmit channel select signal of SIO0	SIO0XFS
GPIO0_3	I/O	GPIO bidirectional data signal, multiplexed with the transmit bit stream clock signal of SIO0	SIO0XCK
GPIO0_4	I/O	GPIO bidirectional data signal, multiplexed with the master clock signal of SIO0	ACKOUT
GPIO0_5	I/O	GPIO bidirectional data signal, multiplexed with CS signal 1 of the static memory interface (SMI)	SMICS1N
GPIO0_6	I/O	GPIO bidirectional data signal, multiplexed with CS signal 1 of the NAND flash	NFCS1N
GPIO0_7	I/O	GPIO bidirectional data signal, multiplexed with the idle indicator signal of the NAND flash	NFRB
GPIO1_0	I/O	GPIO bidirectional data signal, multiplexed with the ready indication signal of the SMI	EBIRDYN
GPIO1_1	I/O	GPIO bidirectional data signal, multiplexed with the ETH collision indicator signal	ECOL
GPIO1_2	I/O	GPIO bidirectional data signal, multiplexed with the ETH carrier sense signal	ECOL
GPIO1_3	I/O	GPIO bidirectional data signal, multiplexed with VI0 horizontal sync signal	VI0HS
GPIO1_4	I/O	GPIO bidirectional data signal, multiplexed with VI0 vertical sync signal	VI0VS
GPIO1_5	I/O	GPIO bidirectional data signal, multiplexed with VI2 horizontal sync signal	VI2HS
GPIO1_6	I/O	GPIO bidirectional data signal, multiplexed with VI2 vertical sync signal	VI2VS
GPIO1_7	I/O	GPIO bidirectional data signal	VOCK
GPIO2_0	I/O	GPIO bidirectional data signal, multiplexed with VO0 SD image output data DAT0 signal	VODAT0
GPIO2_1	I/O	GPIO bidirectional data signal, multiplexed with VO0 SD image output data DAT1 signal	VODAT1

Signal	Direction	Description	Pin
GPIO2_2	I/O	GPIO bidirectional data signal, multiplexed with VO0 SD image output data DAT2 signal	VODAT2
GPIO2_3	I/O	GPIO bidirectional data signal, multiplexed with VO0 SD image output data DAT3 signal	VODAT3
GPIO2_4	I/O	GPIO bidirectional data signal, multiplexed with VO0 SD image output data DAT4 signal	VODAT4
GPIO2_5	I/O	GPIO bidirectional data signal, multiplexed with VO0 SD image output data DAT5 signal	VODAT5
GPIO3_0	I/O	GPIO bidirectional data signal, multiplexed with VI1 DAT0 signal	VI1DAT0
GPIO3_1	I/O	GPIO bidirectional data signal, multiplexed with VI1 DAT1 signal	VI1DAT1
GPIO3_2	I/O	GPIO bidirectional data signal, multiplexed with VI1 DAT2 signal	VI1DAT2
GPIO3_3	I/O	GPIO bidirectional data signal, multiplexed with VI1 DAT3 signal	VI1DAT3
GPIO3_4	I/O	GPIO bidirectional data signal, multiplexed with VI1 DAT4 signal	VI1DAT4
GPIO3_5	I/O	GPIO bidirectional data signal, multiplexed with VI1 DAT5 signal	VI1DAT5
GPIO3_6	I/O	GPIO bidirectional data signal, multiplexed with VI1 DAT6 signal	VI1DAT6
GPIO3_7	I/O	GPIO bidirectional data signal, multiplexed with VI1 DAT7 signal	VI1DAT7
GPIO4_0	I/O	GPIO bidirectional data signal, multiplexed with VI2 DAT0 signal	VI2DAT0
GPIO4_1	I/O	GPIO bidirectional data signal, multiplexed with VI2 DAT1 signal	VI2DAT1
GPIO4_2	I/O	GPIO bidirectional data signal, multiplexed with VI2 DAT2 signal	VI2DAT2
GPIO4_3	I/O	GPIO bidirectional data signal, multiplexed with VI2 DAT3 signal	VI2DAT3
GPIO4_4	I/O	GPIO bidirectional data signal, multiplexed with VI2 DAT4 signal	VI2DAT4
GPIO4_5	I/O	GPIO bidirectional data signal, multiplexed with VI2 DAT5 signal	VI2DAT5
GPIO4_6	I/O	GPIO bidirectional data signal, multiplexed with VI2 DAT6 signal	VI2DAT6

Signal	Direction	Description	Pin
GPIO4_7	I/O	GPIO bidirectional data signal, multiplexed with VI2 DAT7 signal	VI2DAT7
GPIO5_0	I/O	GPIO bidirectional data signal, multiplexed with VI3 DAT0 signal	VI3DAT0
GPIO5_1	I/O	GPIO bidirectional data signal, multiplexed with VI3 DAT1 signal	VI3DAT1
GPIO5_2	I/O	GPIO bidirectional data signal, multiplexed with VI3 DAT2 signal	VI3DAT2
GPIO5_3	I/O	GPIO bidirectional data signal, multiplexed with VI3 DAT3 signal	VI3DAT3
GPIO5_4	I/O	GPIO bidirectional data signal, multiplexed with VI3 DAT4 signal	VI3DAT4
GPIO5_5	I/O	GPIO bidirectional data signal, multiplexed with VI3 DAT5 signal	VI3DAT5
GPIO5_6	I/O	GPIO bidirectional data signal, multiplexed with VI3 DAT6 signal	VI3DAT6
GPIO5_7	I/O	GPIO bidirectional data signal, multiplexed with VI3 DAT7 signal	VI3DAT7

11.5.4 Function Description

Each set of GPIO provides eight programmable input/output pins. Each pin can be configured as input or output. These pins generate the output signals used for specific applications or capture the input signals used for specific applications.

The GPIO can generate maskable interrupts based on the level or transition value. The general purpose input output interrupt (GPIOINTR) signal gives an indication to the interrupt controller, indicating that an interrupt occurs.

11.5.5 Operating Mode

Pin Multiplexing

Each GPIO is multiplexed with other pins. Before using a GPIO pin, you need to configure the IO config register to enable the GPIO pin. For details, see Table 11-21.

Table 11-21 Configuration of pin multiplexing

Signal	Configuration	Multiplexing Description
GPIO0_0	reg37	Multiplexed with the I^2C data signal SDA
GPIO0_1	reg38	Multiplexed with the I^2C clock signal SCL

Signal	Configuration	Multiplexing Description
GPIO0_2	reg39	Multiplexed with the I^2S transmit channel select signal of SIO0
GPIO0_3	reg40	Multiplexed with the transmit bit stream clock signal of SIO0
GPIO0_4	reg41	Multiplexed with the master clock signal of SIO0
GPIO0_5	reg42	Multiplexed with CS signal 1 of the SMI
GPIO0_6	reg43	Multiplexed with CS signal 1 of the NAND flash
GPIO0_7	reg44	Multiplexed with the idle indicator signal of the NAND flash
GPIO1_0	reg45	Multiplexed with the input ready indicator signal and the infrared data receive signal of the SMI
GPIO1_1	reg46	Multiplexed with the ETH collision indicator signal
GPIO1_2	reg47	Multiplexed with the ETH carrier sense signal
GPIO1_3	reg0	Multiplexed with the VI0 horizontal sync signal and the UART2 data receive signal
GPIO1_4	reg1	Multiplexed with the VI0 vertical sync signal and the UART2 data transmit signal
GPIO1_5	reg10	Multiplexed with the VI2 horizontal sync signal and the UART2 data receive signal
GPIO1_6	reg11	Multiplexed with the VI2 vertical sync signal and the UART2 data transmit signal
GPIO1_7	reg28	Multiplexed with the SD image output clock signal, SDIO/MMC clock signal, and SPI clock signal
GPIO2_0	reg29	Multiplexed with the VO SD image output DAT0 signal and SDIO/MMC command signal
GPIO2_1	reg30	Multiplexed with the VO SD image output DAT1 signal, SDIO/MMC DAT0 signal, and SPI output data signal
GPIO2_2	reg31	Multiplexed with the VO SD image output DAT2 signal, SDIO/MMC DAT1 signal, and SPI CS signal 0
GPIO2_3	reg32	Multiplexed with the VO SD image output DAT3 signal, SDIO/MMC DAT2 signal, and SPI CS signal 1
GPIO2_4	reg33	Multiplexed with the VO SD image output DAT4 signal and SDIO/MMC DAT3 signal
GPIO2_5	reg34	Multiplexed with the VO SD image output DAT5 signal and SDIO/MMC card detection signal

Signal	Configuration	Multiplexing Description
GPIO3_0	reg2	Multiplexed with the VI1 DAT0 signal
GPIO3_1	reg3	Multiplexed with the VI1 DAT1 signal
GPIO3_2	reg4	Multiplexed with the VI1 DAT2 signal
GPIO3_3	reg5	Multiplexed with the VI1 DAT3 signal
GPIO3_4	reg6	Multiplexed with the VI1 DAT4 signal
GPIO3_5	reg7	Multiplexed with the VI1 DAT5 signal
GPIO3_6	reg8	Multiplexed with the VI1 DAT6 signal
GPIO3_7	reg9	Multiplexed with the VI1 DAT7 signal
GPIO4_0	reg12	Multiplexed with the VI2 DAT0 signal
GPIO4_1	reg13	Multiplexed with the VI2 DAT1 signal
GPIO4_2	reg14	Multiplexed with the VI2 DAT2 signal
GPIO4_3	reg15	Multiplexed with the VI2 DAT3 signal
GPIO4_4	reg16	Multiplexed with the VI2 DAT4 signal
GPIO4_5	reg17	Multiplexed with the VI2 DAT5 signal
GPIO4_6	reg18	Multiplexed with the VI2 DAT6 signal
GPIO4_7	reg19	Multiplexed with the VI2 DAT7 signal
GPIO5_0	reg20	Multiplexed with the VI3 DAT0 signal
GPIO5_1	reg21	Multiplexed with the VI3 DAT1 signal
GPIO5_2	reg22	Multiplexed with the VI3 DAT2 signal
GPIO5_3	reg23	Multiplexed with the VI3 DAT3 signal
GPIO5_4	reg24	Multiplexed with the VI3 DAT4 signal
GPIO5_5	reg25	Multiplexed with the VI3 DAT5 signal
GPIO5_6	reg26	Multiplexed with the VI3 DAT6 signal
GPIO5_7	reg27	Multiplexed with the VI3 DAT7 signal

Interface Reset

During power-on reset, all registers are cleared, and therefore the pins work as input pins by default.

When the reset signal is valid, the status of the GPIO is as follows:

- The interrupt become invalid after the corresponding bit of GPIO_IE is cleared.
- All registers are cleared.

- All pins are configured as input.
- All raw interrupt registers are cleared.
- The type of the interrupt is configured as edge-sensitive.

General Purpose Input/Output

Each pin can be configured as an input or output pin. To configure a GPIO pin, do as follows:

Step 1 Configure the corresponding bit of the IO config register according to Table 11-21 to enable the required GPIO pin.

Step 2 Configure the GPIO as input or output through the GPIO_DIR register.

- GPIO as an input pin: The external signals are transmitted through the GPIO pin. In this case, the values of the input signals can be viewed through the GPIO_DATA register. Note that the input signals are also transmitted to the pins that are multiplexed with the GPIO pin.
- GPIO as an output pin: The values are written to the GPIO_DATA register and then output through the GPIO. Note that if the GPIO interrupt function is enabled, an interrupt occurs when the output signal meets the triggering condition.

----End

Interrupt Operation

To generate an interrupt and avoid a pseudo interrupt, do as follows:

Step 1 Select the edge-sensitive mode or level-sensitive mode through the GPIO_IS register.

Step 2 Select the falling- or rising-edge-sensitive mode or high- or low-level-sensitive mode through the GPIO_IEV register.

Step 3 If the edge-sensitive mode is selected, you need to select single-edge-sensitive mode or dual-edge-sensitive mode through the GPIO_IBE register.

Step 4 Ensure that the GPIO data lines are stable during the operations.

Step 5 Write 0xFF to the GPIO_IC register to clear the interrupt.

Step 6 Set the GPIO_IE to 1 enable the interrupt.

----End

The GPIO interrupts are controlled through seven registers. When one or multiple 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 so as to enable superior interrupts.
- Level-sensitive mode: The external interrupt source should keep this level until the processor identifies this interrupt.

11.5.6 Register Summary

Table 11-22 lists the base addresses of six sets of GPIO pins.

Table 11-22 Base addresses of the six sets of GPIO pins

Register	Base Address
GPIO0	0x2015_0000
GPIO1	0x2016_0000
GPIO2	0x2017_0000
GPIO3	0x2018_0000
GPIO4	0x2019_0000
GPIO5	0x201A_0000

Table 11-23 lists the offset addresses and descriptions of a single set of internal GPIO registers. Each GPIO from GPIO0 to GPIO5 also has the same set of internal GPIO registers. The address of the GPIOn register is GPIOn base address + offset address of the register.

Table 11-23 Summary of the GPIO registers

Offset Address	Register	Description	Page
0x000–0x3FC	GPIO_DATA	GPIO data register	11-105
0x400	GPIO_DIR	GPIO direction control register	11-107
0x404	GPIO_IS	GPIO interrupt trigger mode register	11-107
0x408	GPIO_IBE	GPIO interrupt edge control register	11-108
0x40C	GPIO_IEV	GPIO interrupt event register	11-108
0x410	GPIO_IE	GPIO interrupt mask register	11-109
0x414	GPIO_RIS	GPIO raw interrupt status register	11-109
0x418	GPIO_MIS	GPIO masked interrupt status register	11-110
0x41C	GPIO_IC	GPIO interrupt clear register	11-110
0x420	GPIO_RESERVED	GPIO reservation register	11-111

11.5.7 Register Description

GPIO_DATA

GPIO_DATA is the 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 configuration of pin multiplexing must be correct). If the bit is configured as input, the system reads the value of the corresponding input pin.

CAUTION

If the corresponding bit of the 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] correspond to GPIO_DATA[7:0] respectively. When the corresponding PADDR bit is high, the system reads or writes the corresponding bit; when the corresponding bit is low, the operation cannot be performed. For example:

- If the address is 0x3FC (0b11_1111_1100), operations on all the eight bits GPIO_DATA[7:0] are valid.
- If the address is 0x200 (0b10_0000_0000), then only the operation on GPIO_DATA[7] is valid.

Offset Address	Register Name	Total Reset Value
0x000–0x3FC	GPIO_DATA	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_data							
Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RW	gpio_data	Input data of the GPIO when the GPIO is configured as input: output data of the GPIO when the GPIO is configured as output. Each bit can be controlled independently. The register is used together with the GPIO_DIR.

GPIO_DIR

GPIO_DIR is the GPIO direction control register. It is used to configure the direction of the GPIO pin.

Offset Address	Register Name	Total Reset Value
0x400	GPIO_DIR	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_dir							
Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RW	gpio_dir	GPIO direction control register. Bit[7:0] correspond to GPIO_DATA[7:0] respectively. Each bit can be controlled independently. 0: The direction of the GPIO pin is input. 1: The direction of the GPIO pin is output.

GPIO_IS

GPIO_IS is the GPIO interrupt trigger mode register. It is used to configure the interrupt trigger mode.

Offset Address	Register Name	Total Reset Value
0x404	GPIO_IS	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_is							

Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RW	gpio_is	GPIO interrupt trigger mode register. Bit[7:0] correspond to GPIO_DATA[7:0] respectively. Each bit is controlled independently. 0: The interrupt is triggered in edge-sensitive mode. 1: The interrupt is triggered in level-sensitive mode.

GPIO_IBE

GPIO_IBE is the GPIO interrupt edge control register. It is used to control the edge where the interrupt occurs.

Offset Address	Register Name	Total Reset Value
0x408	GPIO_IBE	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_ibe							
Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RW	gpio_ibe	GPIO interrupt edge control register. Bit[7:0] correspond to GPIO_DATA[7:0] respectively. Each bit is controlled independently. 0: The interrupt is in single-edge-sensitive mode. The GPIO_IEV controls whether the interrupt is rising-edge-sensitive or falling-edge-sensitive. 1: The interrupt is in dual-edge-sensitive mode.

GPIO_IEV

GPIO_IEV is the GPIO interrupt event register. It is used to configure the interrupt event.

Offset Address	Register Name	Total Reset Value
0x40C	GPIO_IEV	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_iev							

Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RW	gpio_iev	GPIO interrupt event register. Bit[7:0] correspond to GPIO_DATA[7:0] respectively. Each bit is controlled independently. 0: The interrupt is triggered at the falling edge or low level. 1: The interrupt is triggered at the rising edge or high level.

GPIO_IE

GPIO_IE is the GPIO interrupt mask register. It is used to mask the GPIO interrupt.

Offset Address	Register Name	Total Reset Value
0x410	GPIO_IE	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_ie							
Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RW	gpio_ie	GPIO interrupt mask register. Bit[7:0] respectively correspond to GPIO_DATA[7:0]. Each bit is controlled independently. 0: The interrupt is masked. 1: The interrupt is not masked.

GPIO_RIS

GPIO_RIS is the GPIO raw interrupt status register. It is used to query the raw interrupt status.

Offset Address	Register Name	Total Reset Value
0x414	GPIO_RIS	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_ris							

Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RO	gpio_ris	GPIO raw interrupt status register. Bit[7:0] respectively correspond to GPIO_DATA[7:0], indicating the unmasked interrupt status. The status is not under the mask control of the GPIO_IE register. 1: An interrupt occurs. 1: No interrupt occurs.

GPIO_MIS

GPIO_MIS is the GPIO masked interrupt status register. It is used to query the masked interrupt status.

Offset Address	Register Name	Total Reset Value
0x418	GPIO_MIS	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_mis							
Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RO	gpio_mis	GPIO masked interrupt status register. Bit[7:0] respectively correspond to GPIO_DATA[7:0], indicating the masked interrupt status. The status is under the mask control of the GPIO_IE register. 0: The interrupt is invalid. 1: The interrupt is valid.

GPIO_IC

GPIO_IC is the GPIO interrupt clear register. It is used to clear the interrupts generated by the GPIO and clear the GPIO_RIS and GPIO_MIS registers.

Offset Address	Register Name	Total Reset Value
0x41C	GPIO_IC	0x00

Bit	7	6	5	4	3	2	1	0
Name	gpio_ic							

Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	WC	gpio_ic	GPIO interrupt clear register. Bit[7:0] respectively correspond to GPIO_DATA[7:0]. Each bit is controlled independently. 0: The interrupt is not cleared. 1: The interrupt is cleared.

GPIO_RESERVED

GPIO_RESERVED is the GPIO reservation register. It is configured as required.

Offset Address	Register Name	Total Reset Value
0x420	GPIO_RESERVED	0x00

Bit	7	6	5	4	3	2	1	0
Name	reserved							
Reset	0	0	0	0	0	0	0	0

Bits	Access	Name	Description
[7:0]	RW	reserved	These bits must be set to 0x00.

Contents

Figures

Tables

12 Test Interface

12.1 Overview

The Hi3515 provides the joint test action group (JTAG) interfaces that comply with the Institute of Electrical and Electronics Engineers (IEEE) 1149.1 standard. The JTAG interfaces are used for ARM software debugging and board-level test.

12.2 Operating Modes

The Hi3515 provides two operating modes that can be switched from one to anther through the configuration of the TESTMODE pin. Table 12-1 describes the operating modes of the Hi3515.

Table 12-1 Operating modes of the Hi3515

TESTMODE	Description
0	The Hi3515 works properly. In this mode, the JTAG interface can be used for ARM software debugging.
1	The Hi3515 is in test mode. In this mode, the DFT test and board-level test can be performed.

NOTE
DFT = Test for design

12.3 JTAG Debugging

12.3.1 JTAG Interface Signals

The JTAG interface signals of the Hi3515 are the same as standard JTAG interface signals. Table 12-2 describes the JTAG interface signals of the Hi3515.

Table 12-2 JTAG interface signals of the Hi3515

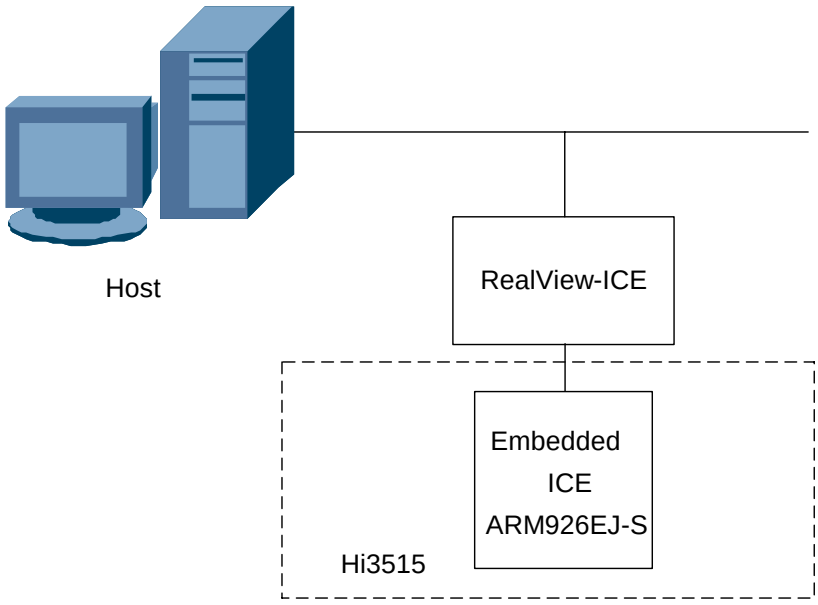
Signal Name	Description
TCK	JTAG clock input, internal pull-down. It is recommended to connect a pull-down resistor.
TDI	JTAG data input, internal pull-up. It is recommended to connect a pull-up resistor to the board.
TMS	JTAG mode select input, internal pull-up. It is recommended to connect a pull-up resistor to the board.
TRSTN	JTAG reset input, internal pull-down. When the Hi3515 works properly, it is recommended to connect a pull-down resistor to the board. When the debugger (such as the Realview-ICE) is connected through the JTAG interface, it is recommended to connect a pull-up resistor to the board.
TDO	JTAG data output. It is recommended to connect a pull-up resistor to the board.

12.3.2 Debugging Mode

Debugging the ARM Software

After an ICE device (such as RealView-ICE) that is compatible with the JTAG interface is connected to the host, the ARM can be debugged through JTAG interfaces by specific debugging software. See Figure 12-1.

Figure 12-1 Diagram of debugging the ARM



When the JTAG interface unit (such as the ARM RealView-ICE) is used for an initial configuration, the TESTMOD pin must be set to 0.

By connecting a pull-up or pull-down resistor to EBIADR22, the system can be connected to ARM926 or the serial advanced technology attachment (SATA) interface separately. For details, see Table 12-3.

CAUTION

- If the JTAG interface of the ARM is not used, it is recommended to connect a pull-down resistor to EBIADR22.
- The pull-up or pull-down change of EBIADR22 takes effect only after the pin is reset.

Table 12-3 Connection setting of ARM926 and SATA interface

EBIADR22	Description
Pull-down	Connect to ARM926EJ-S only
Pull-up	Connect to the SATA interface only

Board-level Test Mode

In addition to software debugging through the JTAG interface, the Hi3515 also supports the board-level interconnection tests, such as the test on the connection between the Hi3515 and other chips. The board-level interconnection test is implemented through a standard JTAG controller. During the board-level interconnection test, TESTMODE must be set to 1.

Contents

13 Video Processing Module

13.1 Video Codec

13.1.1 Overview

The video codec is a video encoding/decoding processing unit that supports the H.264 and JPEG/MJPEG protocols. It consists of a video codec firmware running on an ARM processor and an embedded video encoding/decoding hardware engine. The video codec can encode and decode the videos separately or simultaneously.

13.1.2 Features

The video codec has the following features:

- Supports H.264 Main Profile@Level 4.0 (or lower).
  - In separate video encoding and decoding, up to 4-channel D1 real-time encoding/decoding (PAL: D1@100fps; NTSC: D1@120fps) or 1-channel 720p30 real-time encoding/decoding is supported.
  - In simultaneous video encoding and decoding, up to 2-channle dual stream (D1+CIF) encoding and 2-channel D1 decoding are supported.
- Supports JPEG/MJPEG Baseline encoding/decoding.
  - In separate JPEG encoding or decoding, the frame rate reaches 20 fps when the resolution is 3 megapixels.
  - Simultaneous JPEG encoding and decoding are supported.
- Simultaneously encodes and decodes main and minor streams.
  - The main and minor streams support any combination of the four protocols including H.264/H.264, H.264/JPEG, JPEG/H.264, and JPEG/JPEG.
  - The main and minor streams originate from the same source image. The main stream is obtained directly by encoding the source image. The minor stream is obtained by encoding the downscaled source image.
  - The horizontal and vertical scaling ratios of the encoded images of the main and minor streams can be set to 1:1, 2:1, or 4:1.
  - The encoded image of the minor stream supports the CIF format.
- De-interlaces the interlaced image before encoding.
  - The de-interlace function can be enabled or disabled.
- Supports time-domain filtering before encoding.

  - The time-domain filtering function can be enabled or disabled.
- Supports on screen display (OSD) overlapping before encoding.
  - The OSD overlapping of up to four areas before encoding is supported.
  - Any OSD overlapping within the source image is supported.
  - Up to 129 levels of alpha blending is supported.
  - The OSD overlapping function can be enabled or disabled.
- Supports motion detection.
  - The sum of absolute difference (SAD) can be output.
  - The motion vector (MV) can be output.
- H.264 supports the control of the bit rates including constant bit rate (CBR), variable bit rate (VBR), available bit rate(ABR). The bit rate is controlled within the range of 16 kbit/s to 20 Mbit/s.
- Supports configurable frame rate for encoding.
  - The encoding with low frame rate is supported.
  - Supports the encoding with decimal frame rate.
- Supports the digital watermark technology.

13.2 TDE

13.2.1 Overview

The two-dimensional engine (TDE) draws a graphic through hardware. This greatly reduces the CPU usage and makes full use of the memory bandwidth. The TDE reads and writes the information about bitmap data, scaling filtering coefficients, parameters of linked list nodes, and the linked list through the AXI master bus interface. Through the advanced peripheral bus (APB) interface, the TDE also reads the register configuration information written by the CPU.

The graphic data interface includes two channels: source 1 and source 2. The functions of source 1 and source 2 are as follows:

- Source 1 implements direct copy and direct filling during the single source operation.
- Source 2 implements various complicated operations during the single source operation, such as image scaling and de-flicker output.

Source 1 works with source 2 to implement color blending, raster operations (ROPs), and process the images in microblock format.

13.2.2 Features

The TDE module has the following features:

- Source bitmap 1 supports the format of RGB444, RGB555, RGB565, RGB888, ARGB4444, ARGB1555, ARGB8565, ARGB8888, CLUT1, CLUT2, CLUT4, CLUT8, ACLUT44, ACLUT88, A1, A8, YCbCr888, AYCbCr8888, YCbCr422, byte, half-word, YCbCr400MB, YCbCr422MBH, YCbCr422MBV, YCbCr420MB, or YCbCr444MB. Source bitmap 2 supports the format of RGB444, RGB555, RGB565, RGB888, ARGB4444, ARGB1555, ARGB8565, ARGB8888, CLUT1, CLUT2, CLUT4, CLUT8,

ACLUT44, ACLUT88, A1, A8, YCbCr888, AYCbCr8888, YCbCr422, YCbCr400MB, YCbCr422MBH, YCbCr422MBV, YCbCr420MB, or YCbCr444MB.

- The output bitmap supports the format of RGB444, RGB555, RGB565, RGB888, ARGB4444, ARGB1555, ARGB8565, ARGB8888, CLUT1, CLUT2, CLUT4, CLUT8, ACLUT44, ACLUT88, A1, A8, YCbCr888, AYCbCr8888, YCbCr422, byte, half-word, YCbCr400MB, YCbCr422MBH, YCbCr422MBV, YCbCr420MB, or YCbCr444MB.
- Supports little-endian system only.
- Supports the configurable formats of source bitmap 1, source bitmap 2, and output bitmaps.
- Supports gamma correction and adjustable contrast and luminance.
- Supports color look-up table (CLUT).
- Supports the conversion between RGB and YCbCr.
- Supports direct copy.
- Supports direct filling.
- Supports the 2D-resize operation.
- Supports the deflicker operation.
- Supports the clip operation.
- Supports alpha blending.
- Supports the ROP.
- Supports the color key operation.
- Supports programmable scanning mode.
- Supports clip mask.
- Supports the software interface in synchronous/asynchronous linked list mode.
- Provides the status interrupt.

Contents

A Acronyms and Abbreviations

A

ACD	auto command done
AES	advanced encryption standard
AHB	advanced high-performance bus
AMBA	advanced microcontroller bus architecture
ARM	ARM

B

BVACT	bottom vertical active area
BVBB	bottom vertical back blank
BVFB	bottom vertical front blank

C

CBC	cipher block chaining
CD	command done
CFB	cipher feedback
CL	CAS latency
CPU	central processing unit
CRC	cyclic redundancy check
CRG	clock reset generation
CTR	counter

D

DCRC	data CRC error

DDR	double data-rate
DES	data encryption standard
DFT	design for test
DLL	delay locked loop
DMA	direct memory access
DMAC	direct memory access controller
DQS	data strobe
DRTO	data read timeout
DTO	data transfer over
DVR	digital video recorder
E	
EBE	end-bit error
EBI	external bus interface
ECB	electronic codebook
EOF	end of frame
EOP	end of packet
ETH	Ethernet MAC
F	
FIFO	first in first out
FIQ	fast interrupt request
FRUN	FIFO underrun/overrun error
G	
GPIO	general purpose input/output
H	
HACT	horizontal active area
HCCA	host controller communication area
HFB	horizontal front blank
HLE	hardware locked error
HPW	horizontal pulse width

HTO	data starvation-by-host timeout
HBB	horizontal back blank
I	
I^2C	inter-integrated circuit
IEEE	institute of electrical and electronics engineers
I^2S	inter-IC sound
IR	infrared remoter
IRQ	interrupt request
ISR	interrupt service routine
ITCM	instruction TCM
IV	initialization vector
J	
JTAG	joint test action group
L	
LSB	least significant bit
M	
MAC	media access control
MCU	micro controller unit
MDIO	management data input/output
MII	media independent interface
MMC	multi-media card
MSB	most significant bit
N	
NTSC	national television systems committee
O	
OFB	output feedback
OHCI	open host controller interface

OSD	on screen display
OTG	on-the-go
P	
PAL	phase alternating line
PCB	printed circuit board
PCI	peripheral component interconnect
PCM	pulse code modulation
PID	packet ID
PSRAM	pseudo static random access memory
Q	
QXGA	quantum extended graphics array
R	
RAM	random-access memory
RCRC	response CRC error
RE	response error
ROM	read only memory
ROP	raster operation
RTO	response timeout
RXDR	receive FIFO data request
S	
SAD	sum of absolute difference
SBE	start-bit error
SCL	serial clock
SCR	system clock reference
SD	secure digital
SDA	serial data
SDIO	secure digital input/output
SDRAM	synchronous dynamic random access memory
SFD	start of frame delimiter

SI	specific information
SIO	sonic input/output
SMI	static memory interface
SOF	start of frame
SPI	serial peripheral interface
SRAM	static random access memory
SSP	synchronous serial port

T

TCM	tightly-coupled memory
TDE	two dimension engine
TVACT	top vertical active area
TVBB	top vertical back blank
TVFB	top vertical front blank
TXDR	transmit FIFO data request

U

UART	universal asynchronous receiver transmitter
USB	universal serial bus

V

VACT	vertical active area
VBB	vertical back blank
VBI	vertical blanking interval
VEDU	video encoding decoding unit
VFB	vertical front blank
VIU	video input unit
VOU	video output unit
VPW	vertical pulse width