



Hi3796M V100 Intelligent Network Terminal Media Processor

Data Sheet

Issue 00B03

Date 2015-01-07

Copyright © HiSilicon Technologies Co., Ltd. 2015. All rights reserved.

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

Trademarks and Permissions

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

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

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

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

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
 Bantian, Longgang
 Shenzhen 518129
 People's Republic of China

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

Email: support@hisilicon.com



About This Document

Purpose

This document describes the functions of the intelligent network terminal media processor Hi3796M V100. It also describes how to configure the chip in typical application scenarios.

Related Version

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

Product	Version
Hi3796M	V1XX

Intended Audience

This document is intended for:

- Software development engineers
- Technical support personnel

Symbol Conventions

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

Symbol	Description
DANGER	Alerts you to a high risk hazard that could, if not avoided, result in serious injury or death.
WARNING	Alerts you to a medium or low risk hazard that could, if not avoided, result in moderate or minor injury.



Symbol	Description
CAUTION	Alerts you to a potentially hazardous situation that could, if not avoided, result in equipment damage, data loss, performance deterioration, or unanticipated results.
TIP	Provides a tip that may help you solve a problem or save time.
NOTE	Provides additional information to emphasize or supplement important points in the main text.

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.
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.

Reset Value Conventions

In the register definition tables:

- If the reset value (for the Reset row) of a bit is "?", the reset value is undefined.
- If the reset values of one or multiple bits are "?", the total reset value of a register is undefined and is marked as "-".

Numerical System

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

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



Type	Symbol	Value
	M	1,000,000
	G	1,000,000,000

The expressions of addresses and data are described as follows.

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

Others

Frequencies in this document all comply with the SDH standard. The shortened frequency names and the corresponding nominal frequencies are as follows.

Shortened Frequency Name	Nominal Frequency
19 M	19.44 MHz
38 M	38.88 MHz
77 M	77.76 MHz
622 M	622.08 MHz

Change History

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

Issue 00B03 (2015-01-07)

This issue is the third draft release. Which incorporates the following changes:

Chapter 1, chapter 8, and chapter 11 are modified.

Issue 00B02 (2014-12-16)

This issue is the second draft release. Which incorporates the following changes:



Chapter 3 is modified.

Issue 00B01 (2014-11-17)

This issue is the first draft release.



Contents

1 Overview.....	1-1
1.1 Application Scenarios.....	1-1
1.2 Architecture	1-2
1.2.1 Master Processor.....	1-2
1.2.2 3D Engine	1-3
1.2.3 Security Features.....	1-3
1.2.4 Memory Interfaces.....	1-3
1.2.5 Data Stream Interfaces	1-4
1.2.6 Video CODEC HiVXE	1-4
1.2.7 Graphics and Display Processing.....	1-5
1.2.8 Audio/Video Interfaces	1-5
1.2.9 Peripheral Interfaces	1-6
1.2.10 Boot Modes.....	1-7
1.2.11 Low-Power Control.....	1-7
1.3 Address Space Mapping.....	1-7



Figures

Figure 1-1 Application block diagram of Hi3796M V100 1-2



Tables

Table 1-1 Address space mapping 1-8



1 Overview

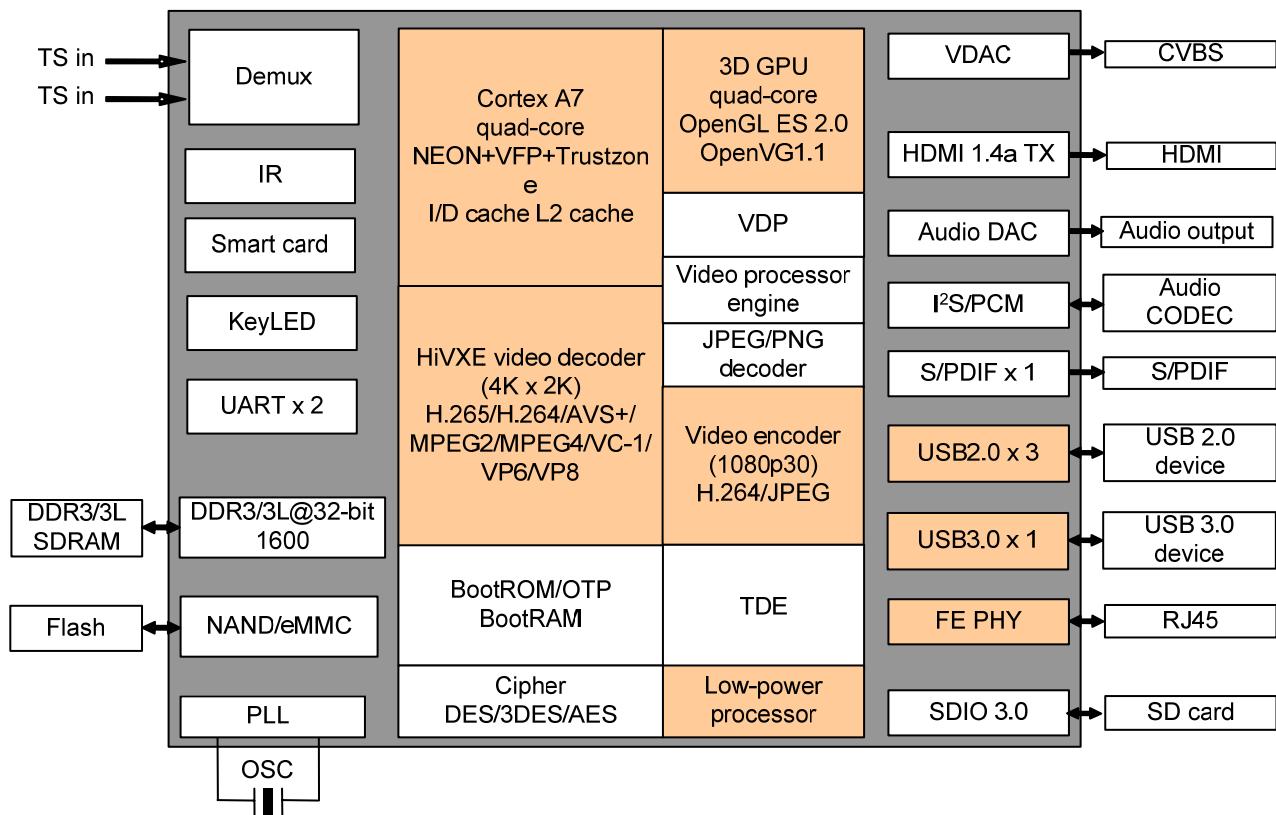
1.1 Application Scenarios

Hi3796M V100 is a cost-effective 4K/H.265 STB chip solution. It brings the best user experience in the industry in terms of stream compatibility, smoothness and picture quality of video playback, and STB performance. With an integrated high-performance quad-core processor and embedded NEON, Hi3796M V100 meets differentiated service requirements. It also supports Dolby and DTS audio processing. To meet the growing requirements on multimedia playback, video communication, and multi-screen transcoding, Hi3796M V100 supports HD video decoding in various formats (including H.265, H.264, AVS+, MPEG2, MPEG4, VC-1, VP6, and VP8) and high-performance H.264 encoding. Hi3796M V100 provides a smooth man-machine interface and rich gaming experience with a high-performance multi-core 2D/3D acceleration engine. It also enables flexible connection schemes with one Ethernet port, three USB 2.0 ports, one USB 3.0 port, and more peripheral interfaces.

[Figure 1-1](#) shows the typical application block diagram of Hi3798M V100.



Figure 1-1 Application block diagram of Hi3796M V100



1.2 Architecture

Hi3796M V100 has the following features:

- Master processor
- 3D engine
- Security processing
- Memory interfaces
- Data stream interfaces
- Video CODEC
- Graphics and display processing
- Audio/Video interfaces
- Peripheral interfaces and boot modes
- Low-power control processor

1.2.1 Master Processor

Hi3796M V100 integrates an ARM Cortex A7 quad-core processor as the master CPU to implement system functions and some audio and video processing tasks. This processor has the following features:



- Maximum frequency of 1.5 GHz
- Independent 32 KB I-cache and 32 KB D-cache, 512 KB shared L2 cache
- Integrated NEON
- Dynamic power consumption reduction such as dynamic voltage and frequency scaling (DVFS) and adaptive voltage scaling (AVS)

1.2.2 3D Engine

Hi3796M V100 integrates the high-performance quad-core Mali GPU to process 3D graphics and videos.

- OpenGL ES 2.0/1.1/1.0
- OpenVG 1.1
- Embedded-system graphics library (EGL)
- 1080p30 UI processing capability
- 1080p30 gaming applications

1.2.3 Security Features

Hi3796M V100 has the following advanced security features:

- Secure boot
- Content protection and encryption and decryption hardware engines compliant with the advanced encryption standard (AES) and triple advanced encryption standard (3DES)
- Joint Test Action Group (JTAG) protection
- One time programmable (OTP)

1.2.4 Memory Interfaces

Hi3796M V100 provides various memory interfaces, including controllers for the DDR SDRAM, NAND flash, and secure digital input/output (SDIO)/multimedia card (MMC).

DDRC

The DDR3/3L SDRAM controller (DDRC) controls the access to the DDR3/3L SDRAM. It supports the following features:

- Maximum 2 GB capacity
- 16-bit or 32-bit memory
- Maximum 800 MHz frequency
- Standby power-down

NANDC

The NAND flash controller (NANDC) controls the access to the asynchronous NAND flash. It supports the following features:

- SLC/MLC flash memory
- Asynchronous NAND flash
- One CS and 8-bit width
- 8-/16-/24-/28-/40-/64-bit error-correcting code (ECC)



- Components with the 2/4/8/16 KB page size
- Randomization for components with the 8/16 KB page size
- Maximum 512 GB capacity

MMC/SD/SDIO Controller

Hi3796M V100 integrates two high-speed large-capacity SDIO 3.0/MMC 4.5 controllers to control the access to the MMC/SD card. In addition, extended peripherals such as Bluetooth and Wi-Fi devices can be connected over the SDIO interface.

One controller supports the 1-/4-bit mode. The other supports the 1-/4-/8-bit mode, and the interface is multiplexed with the NAND flash interface.

1.2.5 Data Stream Interfaces

Hi3796M V100 provides the data stream interfaces including the Ethernet ports and transport stream interface (TSI).

Ethernet Ports

Hi3796M V100 integrates a dual-port Ethernet switching controller, providing two fast Ethernet MAC ports to exchange data with the CPU.

- Working at 10/100 Mbit/s, in half-duplex or full-duplex mode
- One integrated 10/100 Mbit/s PHY and support for energy-efficient Ethernet (EEE)
- Configurable destination MAC address filtering table, which filters the input frames of the Ethernet port
- Traffic control of the CPU interface, protecting the CPU against heavy traffic

TSI

Hi3796M V100 integrates the TSI controller.

- Parsing and demultiplexing of MPEG2 TSs (complying with the ISO 13818-1 GB 17975-1 system layer)
- At most two external DVB serial inputs
- Maximum 96 hardware PID channels

1.2.6 Video CODEC HiVXE

Hi3796M V100 integrates the HD video and graphics CODEC that supports various protocols (H.265/H.264/AVS+/MVC/VC-1/MPEG2/MPEG4/AVS/VP6/VP8/JPEG/PNG), providing powerful video and graphics encoding and decoding capabilities.

- H.265 MP@L5.0
- H.264 BP/MP/HP@L5.1 and MVC decoding
- AVS baseline@L6.0 and AVS+ decoding
- MPEG1 decoding
- MPEG2 MP@HL decoding
- MPEG4 SP@L0–3 and ASP@L0–5 decoding
- MPEG4 short header format (H.263 baseline) decoding



- VC-1 AP@L0–3 decoding
- VP6/VP8 decoding
- Maximum decoding capability of 4K x 2K@30 fps, 1080p@60 fps, or 2x1080p@30 fps
- Full HD JPEG baseline decoding, up to 64 megapixels
- MJPEG baseline decoding, maximum decoding capability of 1080p@40 fps
- PNG decoding, maximum 64 megapixels
- H.264 BP/MP/HP@L4.2 encoding, maximum encoding capability of 1080p@30 fps
- JPEG baseline encoding, maximum encoding capability of 1080p@30 fps
- Variable bit rate (VBR) or constant bit rate (CBR) mode for video encoding

1.2.7 Graphics and Display Processing

Hi3796M V100 integrates the dedicated two dimensional engine (TDE), dedicated hardware composer engine, and dedicated display processing engine.

- Hardware overlaying of multi-channel graphics and video inputs
- Three-layer on-screen display (OSD) and four video layers
- Maximum 4K x 2K image output
- Mosaic and multi-region display
- Mirroring
- 16-bit or 32-bit color depth
- Image or video rotation
- LetterBox and PanScan
- 3D video processing and display
- Multi-order vertical and horizontal scaling of videos and graphics; free scaling
- Low-delay display
- Enhanced full-hardware TDE
- Full-hardware anti-aliasing and anti-flicker
- Color space conversion (CSC) with configurable coefficients
- Image enhancement and denoising
- Deinterlacing
- Sharpening
- Chrominance, luminance, contrast, and saturation adjustment
- Db/Dr processing for graphics and videos

1.2.8 Audio/Video Interfaces

Hi3796M V100 integrates various audio/video input/output interfaces, providing rich audio/video input/output capabilities.

Video Input/Output Interfaces

- One 1920 x 1080p@60 fps output and one SD output from the same source or different sources
- One HDMI 1.4a TX output, maximum 4K x 2K resolution



- One digital-to-analog converter (DAC), supporting one composite video broadcast signal (CVBS) output
- Rovi and vertical blanking interval (VBI) video output

Audio Input/Output Interfaces

- Sony/Philips digital interface format (SPDIF) audio output interface
- One embedded DAC, which supports audio-left and audio-right channels (output interface of the RCA type, low impedance, and imbalance)
- One HDMI TX audio output

1.2.9 Peripheral Interfaces

Hi3796M V100 integrates diverse peripheral interfaces for connecting to various peripherals or extending system functions.

IR

The integrated dedicated infrared remoter (IR) receives data through an IR interface. It has the following features:

- Flexible configuration for decoding data in various formats
- Error check for received data
- IR wakeup
- One input interface

LED/Keypad Controller

Hi3796M V100 integrates the LED/keypad controller for controlling the LED display and key scanning.

USB Controller

Hi3796M V100 integrates three USB 2.0 controllers and supports one USB 3.0 controller.

- All the three USB 2.0 controllers support the host function, and one of them supports the Android debug bridge (ADB) debugging.
- The USB 2.0 ports support the low-speed and high-speed modes and extended hub.

GPIO

Hi3796M V100 integrates multiple groups of GPIO controllers. Each group provides eight programmable input/output pins.

- Each GPIO pin can be configured as an input or output.
- As an input pin, the GPIO can be an interrupt source.
- As an output pin, the GPIO can be independently set to 1 or 0.



UART

Hi3796M V100 integrates two universal asynchronous receiver transmitters (UARTs) for debugging, controlling, or extending external devices such as the Bluetooth and keyboard. One is a 2-wire interface, and the other is a 4-wire interface.

I²C Controller

Hi3796M V100 integrates three inter-integrated circuit (I²C) controllers, which serve as standard master I²C devices to transmit or receive data to or from the slave devices over the I²C bus.

SCI

Hi3796M V100 integrates a smart card interface (SCI) controller and provides one SCI. The SCI controller supports the ISO/IEC 7816-3, ISO/IEC 7816-10, and asynchronous T0, T1, and T4 protocols. The CPU reads data from or writes data to the smart card through the SCI and implements serial-to-parallel conversion (when reading data from the smart card) and parallel-to-serial conversion (when writing data to the smart card).

1.2.10 Boot Modes

Hi3796M V100 supports various boot modes, which can be configured by connecting pull-up or pull-down resistors to the boot pin, providing flexible boot solutions for system applications.

- Booting from the NAND flash
- Booting from the eMMC
- Booting from the SD card

1.2.11 Low-Power Control

Hi3796M V100 supports various low-power modes to dynamically reduce power consumption.

- Various system working modes, including the normal mode, light standby mode, and passive standby mode
- Module low-power control
- DVFS based on CPU load monitoring
- AVS based on CPU timing monitoring
- Ultra-low-power standby design, including various standby wakeup modes such as remote control wakeup and key wakeup

1.3 Address Space Mapping

Hi3796M V100 can be booted from the off-chip NAND flash, eMMC, or SD card.

Hi3796M V100 will boot from the on-chip ROM by default after power-on.



Table 1-1 Address space mapping

Start Address	End Address	Object	Capacity (Byte)	Remarks
0x0000_0000	0x7FFF_FFFF	DRAM	2 GB	-
0x8000_0000	0xF7FF_FFFF	Reserved	1920 MB	
0xF800_0000	0xF800_0FFF	Sysctrl register	4 KB	-
0xF800_1000	0xF800_1FFF	IR register	4 KB	-
0xF800_2000	0xF800_2FFF	Timer 0/Timer 1 register	4 KB	-
0xF800_3000	0xF800_3FFF	LEDC register	4 KB	-
0xF800_4000	0xF800_4FFF	GPIO5 register	4 KB	-
0xF800_5000	0xF800_5FFF	TRNG register	4 KB	-
0xF800_6000	0xF800_6FFF	UART1 register	4 KB	-
0xF800_7000	0xF83F_FFFF	Reserved	4068 KB	-
0xF840_0000	0xF840_FFFF	8051 Local RAM	64 KB	-
0xF841_0000	0xF89F_FFFF	Reserved	6080 KB	
0xF8A0_0000	0xF8A0_1FFF	A7MP peripheral register	8 KB	-
0xF8A0_2000	0xF8A1_FFFF	Reserved	120 KB	
0xF8A2_0000	0xF8A2_0FFF	PERI_CTRL register	4 KB	-
0xF8A2_1000	0xF8A2_1FFF	IO_CONFIG register	4 KB	-
0xF8A2_2000	0xF8A2_2FFF	CRG register	4 KB	-
0xF8A2_3000	0xF8A2_3FFF	PMC register	4 KB	-
0xF8A2_4000	0xF8A2_8FFF	Reserved	20 KB	
0xF8A2_9000	0xF8A2_9FFF	Timer 2 and timer 3 register	4 KB	-
0xF8A2_A000	0xF8A2_AFFF	Timer 4 and timer 5 register	4 KB	-
0xF8A2_B000	0xF8A2_BFFF	Timer 6 and timer 7 register	4 KB	-
0xF8A2_C000	0xF8A2_CFFF	WDG0 register	4 KB	-
0xF8A2_D000	0xF8A2_FFFF	Reserved	12 KB	
0xF8A3_0000	0xF8A3_FFFF	MDDRC0 register	64 KB	-
0xF8A4_0000	0xF8A4_FFFF	Reserved	64 KB	
0xF8A5_0000	0xF8A5_FFFF	DDR_TEST register	64 KB	-
0xF8A6_0000	0xF8A7_FFFF	Reserved	128 KB	
0xF8A8_0000	0xF8A8_0FFF	SEC_CFG register	4 KB	-
0xF8A8_1000	0xF8A8_FFFF	Reserved	60 KB	



Start Address	End Address	Object	Capacity (Byte)	Remarks
0xF8A9_0000	0xF8A9_FFFF	MKL	64 KB	-
0xF8AA_0000	0xF8AA_FFFF	Reserved	64 KB	
0xF8AB_0000	0xF8AB_FFFF	OTP register	64 KB	-
0xF8AC_0000	0xF8AF_FFFF	Reserved	256 KB	
0xF8B0_0000	0xF8B0_0FFF	UART0 register	4 KB	-
0xF8B0_1000	0xF8B0_FFFF	Reserved	60 KB	-
0xF8B1_0000	0xF8B1_0FFF	I ² C0	4 KB	-
0xF8B1_1000	0xF8B1_1FFF	I ² C1	4 KB	-
0xF8B1_2000	0xF8B1_2FFF	I ² C2	4 KB	-
0xF8B1_3000	0xF8B1_7FFF	Reserved	20 KB	
0xF8B1_8000	0xF8B1_8FFF	SCI0 register	4 KB	-
0xF8B1_9000	0xF8B1_FFFF	Reserved	28 KB	
0xF8B2_0000	0xF8B2_0FFF	GPIO0 register	4 KB	-
0xF8B2_1000	0xF8B2_1FFF	GPIO1 register	4 KB	-
0xF8B2_2000	0xF8B2_2FFF	GPIO2 register	4 KB	-
0xF8B2_3000	0xF8B2_3FFF	GPIO3 register	4 KB	-
0xF8B2_4000	0xF8B2_4FFF	GPIO4 register	4 KB	-
0xF8B2_5000	0xF8B2_5FFF	Reserved	4 KB	
0xF8B2_6000	0xF8B2_6FFF	GPIO6 register	4 KB	-
0xF8B2_7000	0xF8C0_FFFF	Reserved	932 KB	
0xF8C1_0000	0xF8C1_FFFF	TDE register	64 KB	-
0xF8C2_0000	0xF8C2_FFFF	Reserved	64 KB	
0xF8C3_0000	0xF8C3_FFFF	VDH register	64 KB	-
0xF8C4_0000	0xF8C4_FFFF	JPGD0 register	64 KB	-
0xF8C5_0000	0xF8C5_FFFF	Reserved	64 KB	-
0xF8C6_0000	0xF8C6_FFFF	BPD register	64 KB	-
0xF8C7_0000	0xF8C7_FFFF	PGD register	64 KB	-
0xF8C8_0000	0xF8C8_FFFF	Video encoding/decoding unit (VEDU) register	64 KB	-
0xF8C9_0000	0xF8C9_FFFF	JPEG encoder (JPGE) register	64 KB	-
0xF8CA_0000	0xF8CA_FFFF	Reserved	64 KB	-



Start Address	End Address	Object	Capacity (Byte)	Remarks
0xF8CB_0000	0xF8CB_FFFF	VPSS0 register	64 KB	-
0xF8CC_0000	0xF8CC_FFFF	Video display processor (VDP) register	64 KB	-
0xF8CD_0000	0xF8CD_FFFF	AIAO register	64 KB	-
0xF8CE_0000	0xF8CE_FFFF	HDMI_TX register	64 KB	-
0xF8CF_0000	0xF91F_FFFF	Reserved	5184 KB	-
0xF920_0000	0xF923_FFFF	GPU register	256 KB	-
0xF924_0000	0xF980_FFFF	Reserved	5952 KB	-
0xF981_0000	0xF981_FFFF	NANDC register	64 KB	-
0xF982_0000	0xF982_FFFF	SDIO0 REG	64 KB	-
0xF983_0000	0xF983_FFFF	SDIO1 REG	64 KB	-
0xF984_0000	0xF984_FFFF	SF0 register	64 KB	-
0xF985_0000	0xF986_FFFF	Reserved	128 KB	-
0xF987_0000	0xF987_FFFF	Direct memory access controller (DMAC) register	64 KB	-
0xF988_0000	0xF988_FFFF	USB2Host0 OHCI register	64 KB	-
0xF989_0000	0xF989_FFFF	USB2Host0 EHCI register	64 KB	-
0xF98A_0000	0xF98A_FFFF	USB3_0 register	64 KB	-
0xF98B_0000	0xF98B_FFFF	Reserved	64 KB	-
0xF98C_0000	0xF98F_FFFF	USB2OTG0 register	256 KB	-
0xF990_0000	0xF991F_FFFF	Reserved	128 KB	-
0xF992_0000	0xF992_FFFF	USB2Host1 OHCI register	64 KB	-
0xF993_0000	0xF993_FFFF	USB2Host1 EHCI register	64 KB	-
0xF994_0000	0xF99F_FFFF	Reserved	704 KB	-
0xF9A0_0000	0xF9A0_FFFF	MutiCIPHER register	64 KB	-
0xF9A1_0000	0xF9A1_FFFF	SHA	64 KB	-
0xF9A2_0000	0xF9BF_FFFF	Reserved	1920 KB	-
0xF9C0_0000	0xF9C0_FFFF	PVR register	64 KB	-
0xF9C1_0000	0xFDFF_FFFF	Reserved	69568 KB	-
0xFE00_0000	0xFE0F_FFFF	NAND flash	1 MB	-
0xFE10_0000	0xFFFFD_FFFF	Reserved	31616 KB	-



Start Address	End Address	Object	Capacity (Byte)	Remarks
0xFFFFE_0000	0xFFFFE_FFFF	BOOTRAM (in remap mode) or reserved (in remap clear mode)	64 KB	-
0xFFFFF_0000	0xFFFFF_FFFF	BOOTROM (in remap mode) or BOOTRAM (in remap clear mode)	64 KB	-



Contents

2 Security Subsystem.....	2-1
2.1 Features	2-1



2 Security Subsystem

2.1 Features

The security subsystem has the following security features:

- Secure boot. After hardware is reset, the chip boots from the embedded on-chip ROM and then from the flash memory. In addition, signature authentication by using the boot loader is supported.
- Control word (CW) protection by using hardware
- Content protection and AES and 3DES algorithms
- JTAG protection
- Trustzone feature
- OTP and unique ID



Contents

3 System	3-1
3.1 Processor Subsystem.....	3-1
3.1.1 ARM Cortex A7 Processor.....	3-1
3.1.2 L2 Cache.....	3-1
3.2 Reset.....	3-2
3.2.1 Overview.....	3-2
3.2.2 Reset Control	3-2
3.2.3 Reset Configuration	3-3
3.3 Clock	3-8
3.3.1 Overview.....	3-8
3.3.2 Clock Control Block Diagram	3-8
3.3.3 Clock Configurations	3-9
3.3.4 Register Summary.....	3-23
3.3.5 Register Description.....	3-25
3.4 System Controller.....	3-70
3.4.1 Overview.....	3-70
3.4.2 Features.....	3-70
3.4.3 Function Description.....	3-70
3.4.4 Register Summary.....	3-72
3.4.5 Register Description.....	3-75
3.5 Peripheral Controller.....	3-107
3.5.1 Overview.....	3-107
3.5.2 Features.....	3-107
3.5.3 Register Summary.....	3-108
3.5.4 Register Description.....	3-109
3.6 Interrupt System	3-142
3.7 Timer	3-149
3.7.1 Overview.....	3-149
3.7.2 Features.....	3-149
3.7.3 Function Description.....	3-149
3.7.4 Working Process	3-150
3.7.5 Register Summary.....	3-151



3.7.6 Register Description.....	3-151
3.8 DMA Controller	3-159
3.8.1 Overview.....	3-159
3.8.2 Features.....	3-160
3.8.3 Function Description.....	3-160
3.8.4 Operating Mode	3-163
3.8.5 Register Summary.....	3-165
3.8.6 Register Description.....	3-166



Figures

Figure 3-1 Block diagram for controlling reset signals	3-2
Figure 3-2 Functional block diagram of the clock management module	3-9
Figure 3-3 Functional block diagram of the DMAC	3-161
Figure 3-4 Updating channel registers by using LLI	3-162
Figure 3-5 Connections between the DMAC and other modules	3-163
Figure 3-6 Structure of the linked list for the DMAC	3-175



Tables

Table 3-1 Reset signal types	3-3
Table 3-2 Soft reset configurations	3-4
Table 3-3 Frequency configurations of NANDC clocks.....	3-10
Table 3-4 Frequency configuration of DDR clocks	3-10
Table 3-5 Frequency configurations of UART clocks	3-10
Table 3-6 Frequency configurations of I ² C clocks.....	3-11
Table 3-7 Frequency configuration of the SSP clock.....	3-11
Table 3-8 Frequency configuration of the SCI clock.....	3-11
Table 3-9 Frequency configurations of VDH clocks	3-11
Table 3-10 Frequency configurations of JPGD clocks	3-12
Table 3-11 Frequency configuration of the PGD clock	3-12
Table 3-12 Frequency configuration of the BPD clock	3-12
Table 3-13 Frequency configurations of VENC clocks	3-12
Table 3-14 Frequency configuration of the JPGE clock	3-13
Table 3-15 Frequency configurations of TDE clocks	3-13
Table 3-16 Frequency configurations of SDIO0 clocks.....	3-13
Table 3-17 Frequency configurations of SDIO1 clocks.....	3-14
Table 3-18 Frequency configuration of the DMA clock	3-14
Table 3-19 Frequency configurations of USB3CTRL clocks	3-14
Table 3-20 Frequency configurations of USB2 CTRL0 clocks	3-15
Table 3-21 Frequency configurations of USB2 PHY0 clocks	3-15
Table 3-22 Frequency configuration of the CA clock	3-16
Table 3-23 Frequency configuration of the SHA clock	3-16
Table 3-24 Frequency configuration of the PMC clock.....	3-16
Table 3-25 Frequency configurations of ETH clocks	3-16
Table 3-26 Frequency configurations of GPU clocks.....	3-17



Table 3-27 Frequency configurations of VDP clocks	3-17
Table 3-28 Frequency configurations of VPSS clocks	3-18
Table 3-29 Frequency configurations of PVR-1 clocks.....	3-18
Table 3-30 Frequency configuration of the PVR-2 clock	3-19
Table 3-31 Frequency configurations of HDMI TX CTRL clocks	3-19
Table 3-32 Frequency configuration of the HDMI TX PHY clock.....	3-20
Table 3-33 Frequency configuration of the ADAC clock	3-20
Table 3-34 Frequency configurations of AIAO clocks	3-20
Table 3-35 Frequency configurations of the VDAC clocks.....	3-20
Table 3-36 Frequency configurations of FE PHY clocks	3-21
Table 3-37 Frequency configuration of the GPU clock	3-21
Table 3-38 Frequency configuration of the DDR clock.....	3-21
Table 3-39 Frequency configurations of WDG clocks	3-21
Table 3-40 Frequency configurations of USB2 PHY1 clocks	3-22
Table 3-41 Frequency configurations of USB3 PHY clocks	3-22
Table 3-42 Frequency configurations of USB2 CTRL1 clocks	3-22
Table 3-43 Summary of CRG registers (base address: 0xF8A2_2000)	3-23
Table 3-44 Summary of SYS_CTRL registers (base address: 0xF800_0000).....	3-72
Table 3-45 Summary of PERI_CTRL registers (base address: 0xF8A2_0000).....	3-108
Table 3-46 Interrupt sources of Cortex A7	3-142
Table 3-47 Interrupt sources of the MCU	3-147
Table 3-48 Summary of timer registers	3-151
Table 3-49 Summary of DMAC registers (base address: 0xF987_0000)	3-165
Table 3-50 Mapping between the value of DBSize or SBSIZE and the burst size.....	3-179
Table 3-51 Mapping between DWidth or SWidth and the transfer bit width.....	3-179
Table 3-52 Definitions of the prot field of DMAC_CX_CONTROL.....	3-180
Table 3-53 Flow controllers and transfer types corresponding to the flow_cntrl field	3-183



3 System

3.1 Processor Subsystem

3.1.1 ARM Cortex A7 Processor

Hi3796M V100 adopts the ARM Cortex-A7 quad-core processor, which has the following features:

- Maximum 1.5 GHz, 11400 DMIPS working frequency
- 32 KB I-cache and 32 KB D-cache (supporting the read-allocate mode) for the L1 cache
- Integrated NEON with the floating-point unit (FPU) coprocessor
- Management unit (MMU)
- Integrated generic interrupt controller (GIC), which supports 160 interrupt sources
- DVFS and AVS
- Support for OSs such as Linux and Android
- JTAG debugging interface
- Integrated L2 cache

3.1.2 L2 Cache

The integrated L2 cache in the Cortex-A7 processor has the following features:

- Provides the 512 KB capacity.
- Works based on the physical address and physical tag.
- Provides the 64-byte (16 words or 512 bits) cache line and supports the write operation by byte.
- Supports the 8-way set-associative structure.
- Supports write-back, read-allocate, and write-allocate operations. The actual mode depends on the page table.
- Supports the forcible read-allocate function, which forces all cacheable write operations to be transparently transmitted to the downstream module.
- Supports enable/disable of the L1 cache and L2 cache at the same time. If the caches are disabled, they are bypassed, and all the operations on L1 are transferred to the DDR memory transparently.
- Supports the read-allocate function for the L2 D-cache.

- Supports the atomic exclusive operation but not atomic lock operation.
- Supports clean and invalidate operations by way, way+index, or address.
- Supports prefetching of instructions and data and configurable prefetch mode.

3.2 Reset

3.2.1 Overview

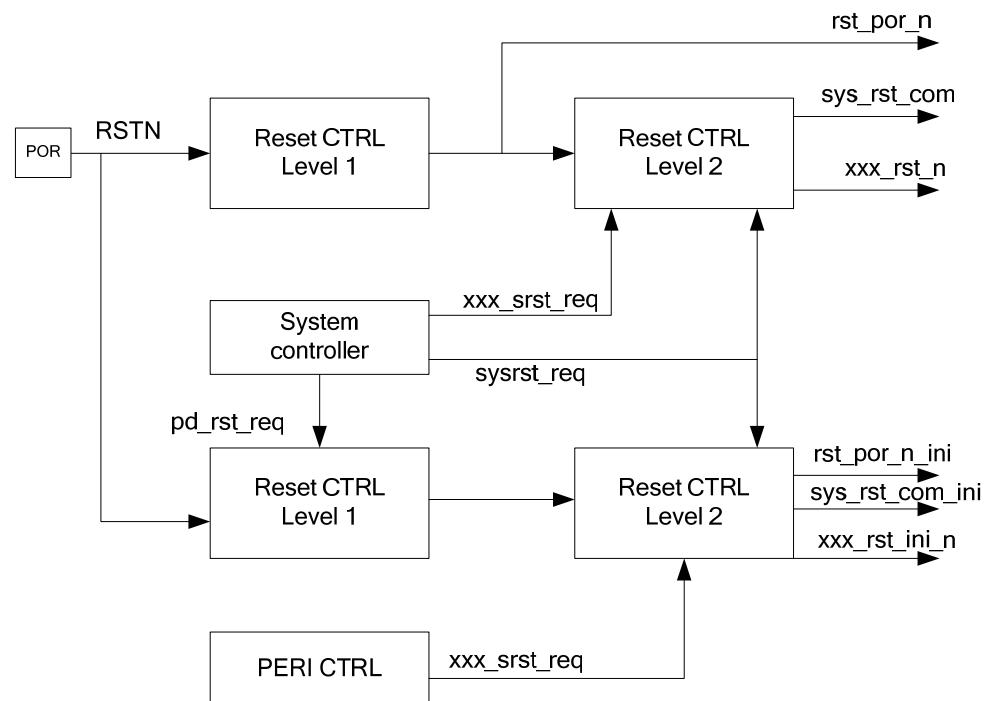
The reset management module resets the entire chip and all functional modules in a unified manner as follows:

- Manages and controls power-on reset (POR).
- Controls the system soft reset and separate soft reset of each functional module.
- Controls the soft reset of power-off areas in low-power mode.
- Synchronizes reset signals to the clock domain of each module.
- Generates reset signals for each internal functional module.

3.2.2 Reset Control

Figure 3-1 shows the block diagram for controlling reset signals.

Figure 3-1 Block diagram for controlling reset signals



RSTN: POR signal. It is the output signal of the POR module in the chip.

sysrst_req: system soft reset request signal. It is derived from the system controller.

pd_rst_req: reset request signal for power-off areas. It is derived from the system controller.

xxx_srst_req: separate soft reset request signal of each module. It is derived from the system controller or



CRG controller, for example, tde_srst_req.

[Table 3-1](#) describes the output reset signals.

Table 3-1 Reset signal types

Reset Signal Type	Generation Mode	Purpose
rst_por_n	Generated by dejittering and synchronizing the RSTN signal	Resets the system controller.
sys_RST_com	Generated by logically combining the POR signal with the system soft reset signal	Resets the modules in the always-on area that support global soft reset but not separate soft reset.
xx_RST_n	Generated by combining the system reset signal sys_RST_com and the separate soft reset signal of each module, for resetting IP modules in the always-on area	Resets the modules that support separate soft reset in the always-on area. Those modules are reset upon POR, global soft reset, or separate soft reset.
rst_por_n_ini	Generated by dejittering and synchronizing the RSTN signal	Resets the A7 and NANDC in the power-off area.
sys_RST_com_ini	Generated by logically combining a POR signal with a system soft reset signal	Resets the modules in the power-off area that support global soft reset but not separate soft reset.
xx_RST_ini_n	Generated by combining the system reset signal sys_RST_com_ini and the separate soft reset signal of each module, for resetting IP modules in the power-off area	Resets the modules that support separate soft reset in the power-off area. Those modules are reset upon POR, global soft reset, or separate soft reset.

3.2.3 Reset Configuration

POR

RSTN is the output signal of the POR module in the HD chip. POR is implemented only when all of the following conditions are met:

- The POR module outputs a low-level pulse.
- The input clock from the crystal oscillator clock input pin XIN is correct.

System Reset

The system is reset in either of the following ways:

- POR
- System soft reset, which is controlled by the system controller [SC_SYSRES](#)



Soft Reset

Soft reset is controlled by configuring the corresponding system controller. For details, see [Table 3-2](#).



CAUTION

- After a system soft reset request is sent, the circuit reset can be deasserted only after 20 system clock cycles.
- The separate soft reset of each module cannot be automatically deasserted. That is, each module is reset when a certain bit is set to 1, and the module reset is deasserted only when the bit is set to 0.
- During power-on, the USB 2.0 PHY and the controller are reset by default. They start to work properly only when the reset is deasserted.

Table 3-2 Soft reset configurations

Signal	Description
sysrst_req	System soft reset request, valid at the rising edge. When the control bit is changed from 0 to 1, a valid reset is executed. Then this reset is automatically deasserted in the chip. This signal can be controlled by configuring SC_SYSRES .
m8051_srst_req	Soft reset control for the microprogrammed control unit (MCU). This signal can be controlled by configuring SC_CLKGATE_SRST_CTRL bit[1].
ir_srst_req	Soft reset control for the IR module. This signal can be controlled by configuring SC_CLKGATE_SRST_CTRL bit[5].
timer_srst_req	Soft reset control for the timer module. This signal can be controlled by configuring SC_CLKGATE_SRST_CTRL bit[7].
led_srst_req	Soft reset control for the light emitting diode control (LEDCCTRL) module. This signal can be controlled by configuring SC_CLKGATE_SRST_CTRL bit[9].
uart_srst_req	Soft reset control for the UART module. This signal can be controlled by configuring SC_CLKGATE_SRST_CTRL bit[13].
pd_RST_req	Reset control for the power-off area. This signal can be controlled by configuring SC_CLKGATE_SRST_CTRL bit[28].
nf_srst_req	Soft reset control for the NANDC. This signal can be controlled by configuring PERI_CRG24 bit[4].
hipack_srst_req	Soft reset control for the HIPACK module. This signal can be controlled by configuring PERI_CRG25 bit[8].



Signal	Description
ddrc_srst_req	Soft reset control for the DDRC. This signal can be controlled by configuring PERI_CRG25 bit[9].
ddrt_srst_req	Soft reset control for the double data rate test (DDRT) module. This signal can be controlled by configuring PERI_CRG25 bit[10].
uart1_srst_req	Soft reset control for UART1. This signal can be controlled by configuring PERI_CRG26 bit[1].
uart2_srst_req	Soft reset control for UART2. This signal can be controlled by configuring PERI_CRG26 bit[5].
i2c0_srst_req	Soft reset control for I ² C0. This signal can be controlled by configuring PERI_CRG27 bit[5].
i2c1_srst_req	Soft reset control for I ² C1. This signal can be controlled by configuring PERI_CRG27 bit[9].
i2c2_srst_req	Soft reset control for I ² C2. This signal can be controlled by configuring PERI_CRG27 bit[13].
ssp0_srst_req	Soft reset control for SSP0. This signal can be controlled by configuring PERI_CRG28 bit[1].
sci0_srst_req	Soft reset control for SCI0. This signal can be controlled by configuring PERI_CRG29 bit[1].
vdh_srst_req	Soft reset control for the video decoder for high-definition (VDH). This signal can be controlled by configuring PERI_CRG30 bit[4].
jpgd_srst_req	Soft reset control for JPGD0. This signal can be controlled by configuring PERI_CRG31 bit[4].
pgd_srst_req	Soft reset control for the PNG and GIF decoder (PGD). This signal can be controlled by configuring PERI_CRG33 bit[4].
bpd_srst_req	Soft reset control for the bit plan decoder (BPD). This signal can be controlled by configuring PERI_CRG34 bit[4].
venc_srst_req	Soft reset control for the video encoder (VENC). This signal can be controlled by configuring PERI_CRG35 bit[4].
jpge_srst_req	Soft reset control for the JPEG encoder (JPGE). This signal can be controlled by configuring PERI_CRG36 bit[4].
tde_srst_req	Soft reset control for the two-dimensional engine (TDE). This signal can be controlled by configuring PERI_CRG37 bit[4].
sdio0_srst_req	Soft reset control for SDIO0. This signal can be controlled by configuring PERI_CRG39 bit[4].
sdio1_srst_req	Soft reset control for SDIO1. This signal can be controlled by configuring PERI_CRG40 bit[4].



Signal	Description
dma_srst_req	Soft reset control for the direct memory access (DMA) module. This signal can be controlled by configuring PERI_CRG41 bit[4].
usb3_vcc_srst_req	Soft reset control for USB3CTRL VCC. This signal can be controlled by configuring PERI_CRG44 bit[12].
usb2_bus_srst_req	Soft reset control for the USB2 CTRL0 bus. This signal can be controlled by configuring PERI_CRG46 bit[12].
usb2_utmi0_srst_req	Soft reset control for USB2 CTRL0 USB 2.0 transceiver macrocell interface 0 (UTMI0). This signal can be controlled by configuring PERI_CRG46 bit[13].
usb2_utmi1_srst_req	Soft reset control for USB2 CTRL0 UTMI1. This signal can be controlled by configuring PERI_CRG46 bit[14].
usb2_hst_phy_srst_req	Soft reset control for the USB2 CTRL0 HST_PHY. This signal can be controlled by configuring PERI_CRG46 bit[16].
usb2_otg_phy_srst_req	Soft reset control for the USB2 CTRL0 OTG_PHY. This signal can be controlled by configuring PERI_CRG46 bit[17].
usb2_phy_srst_req	Soft reset control for the USB2 PHY0 port. This signal can be controlled by configuring PERI_CRG47 bit[8].
usb2_phy_srst_treq0	Soft reset control for the USB2 PHY0 TPORT. This signal can be controlled by configuring PERI_CRG47 bit[9].
usb2_phy_srst_treq1	Soft reset control for the USB2 PHY1 TPORT. This signal can be controlled by configuring PERI_CRG47 bit[11].
otp_srst_req	Soft reset control for the OTP module. This signal can be controlled by configuring PERI_CRG48 bit[10].
ca_ci_srst_req	Soft reset control for the cipher. This signal can be controlled by configuring PERI_CRG48 bit[9].
sha_srst_req	Soft reset control for the secure hash algorithm (SHA) module. This signal can be controlled by configuring PERI_CRG49 bit[4].
pmc_srst_req	Soft reset control for the power management control (PMC) module. This signal can be controlled by configuring PERI_CRG50 bit[4].
hrst_eth_s	Soft reset control for the ETH module. This signal can be controlled by configuring PERI_CRG51 bit[4].
gpu_srst_req	Soft reset control for the GPU. This signal can be controlled by configuring PERI_CRG53 bit[4].
vou_srst_req	Soft reset control for the video output unit (VOU). This signal can be controlled by configuring PERI_CRG54 bit[30].



Signal	Description
vpss_srst_req	Soft reset control for the video processing subsystem (VPSS). This signal can be controlled by configuring PERI_CRG60 bit[4].
pvr_srst_req	Soft reset control for the personal video recorder (PVR). This signal can be controlled by configuring PERI_CRG63 bit[22].
hdmitx_ctrl_bus_srst_req	Soft reset control for the HDMITX_CTRL bus. This signal can be controlled by configuring PERI_CRG67 bit[8].
hdmitx_ctrl_srst_req	Soft reset control for HDMITX_CTRL. This signal can be controlled by configuring PERI_CRG67 bit[9].
hdmitx_phy_srst_req	Soft reset control for the HDMI TX PHY. This signal can be controlled by configuring PERI_CRG68 bit[4].
adac_srst_req	Soft reset control for the audio digital-to-analog converter (ADAC). This signal can be controlled by configuring PERI_CRG69 bit[4].
aiao_srst_req	Soft reset control for the audio input/audio output (AIAO) module. This signal can be controlled by configuring PERI_CRG70 bit[4].
fephy_srst_req	Soft reset control for the FE PHY. This signal can be controlled by configuring PERI_CRG72 bit[4].
wdg0_srst_req	Soft reset control for WDG0. This signal can be controlled by configuring PERI_CRG94 bit[4].
usb2_phy1_srst_req	Soft reset control for the USB2 PHY1 port. This signal can be controlled by configuring PERI_CRG100 bit[8].
usb2_phy1_srst_treq	Soft reset control for the USB2 PHY1 TPORT. This signal can be controlled by configuring PERI_CRG100 bit[9].
usb3_phy_srst_req	Soft reset control for the USB3 PHY port. This signal can be controlled by configuring PERI_CRG101 bit[4].
usb3_phy_srst_treq	Soft reset control for the USB3 PHY TPORT. This signal can be controlled by configuring PERI_CRG101 bit[5].
usb2_bus_srst_req1	Soft reset control for the USB2 CTRL1 bus. This signal can be controlled by configuring PERI_CRG102 bit[12].
usb2_utmi0_srst_req1	Soft reset control for USB2 CTRL1 UTMIO. This signal can be controlled by configuring PERI_CRG102 bit[14].
usb2_hst_phy_srst_req1	Soft reset control for the USB2 CTRL1 HST_PHY. This signal can be controlled by configuring PERI_CRG102 bit[16].

3.3 Clock

3.3.1 Overview

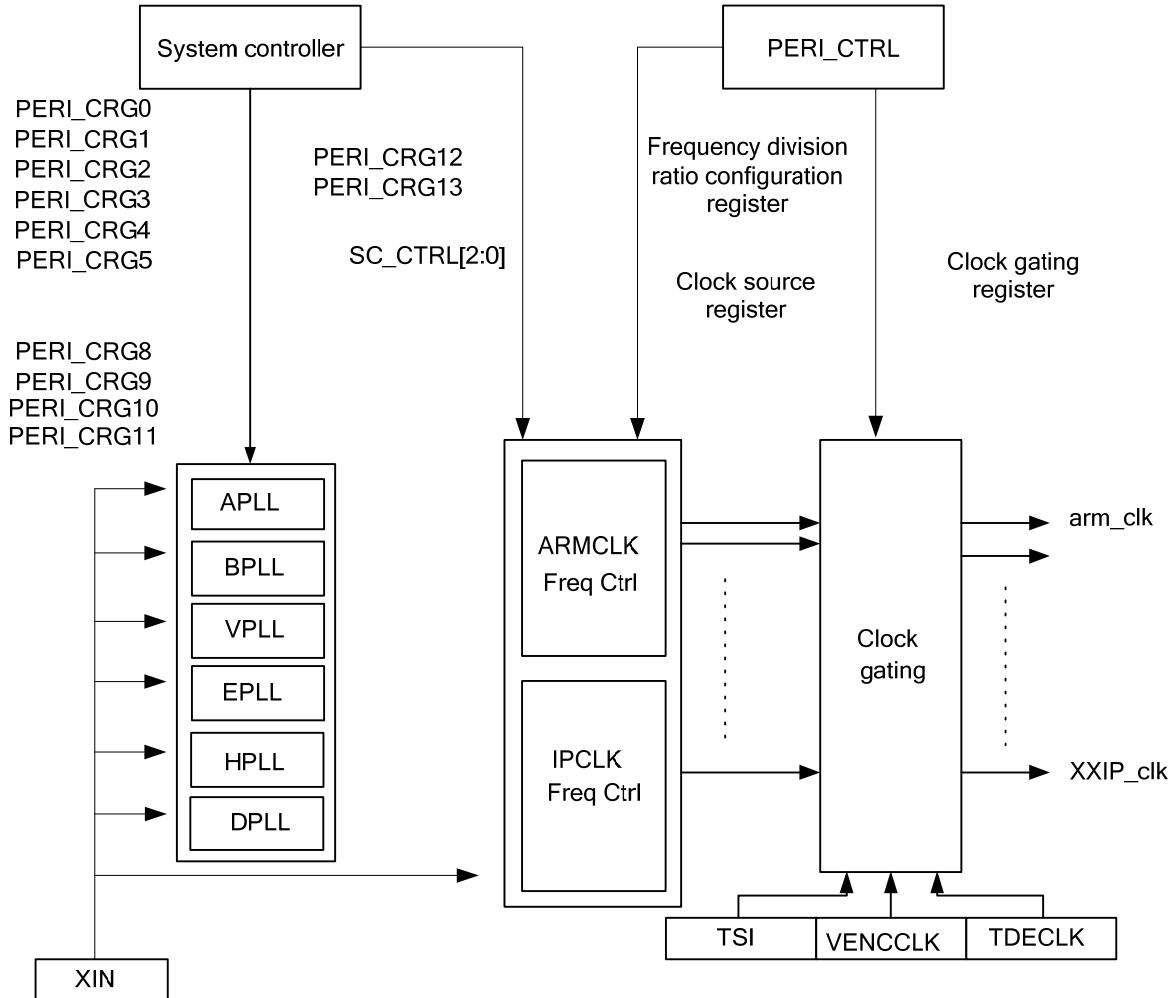
The clock management module provides the following functions:

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

3.3.2 Clock Control Block Diagram

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

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



The inputs of the clock management module include:

- Clock inputs from pins
- Clock control inputs from the system controller



The clock management module consists of the following three parts:

- PLLs including the APLL, BPLL, VPLL, EPLL, DPLL, and HPLL that generate the required clocks with different frequencies for the system
- ARM frequency control unit (ARMCLK Freq Ctrl) and module clock frequency control unit (IPCLK Freq Ctrl)
- Clock gating management unit

3.3.3 Clock Configurations

3.3.3.1 Configuring the NANDC Clock Frequency

[Table 3-3](#) describes frequency configurations of NANDC clocks.

Table 3-3 Frequency configurations of NANDC clocks

Signal	Description
nf_clk_sel	NANDC clock select. This signal can be controlled by configuring PERI_CRG24 bit[10:8].
nf_cken	NANDC clock state. This signal can be controlled by configuring PERI_CRG24 bit[0].

3.3.3.2 Configuring the DDR Clock Frequency

[Table 3-4](#) describes frequency configurations of DDR clocks.

Table 3-4 Frequency configuration of DDR clocks

Signal	Description
ddrt_cken	DDRT clock status. This signal can be controlled by configuring PERI_CRG25 bit[12].
ddrc_cken	DDRC clock status. This signal can be controlled by configuring PERI_CRG25 bit[3].
hipack_cken	HIPACK clock status. This signal can be controlled by configuring PERI_CRG25 bit[2].
ddrphy_cken	DDR PHY clock status. This signal can be controlled by configuring PERI_CRG25 bit[1].
ddr_cken	DDR clock status. This signal can be controlled by configuring PERI_CRG25 bit[0].

3.3.3.3 Configuring the UART Clock Frequency

[Table 3-5](#) describes frequency configurations of UART clocks.



Table 3-5 Frequency configurations of UART clocks

Signal	Description
uart1_cken	UART1 clock status. This signal can be controlled by configuring PERI_CRG26 bit[0].
uart2_cken	UART2 clock status. This signal can be controlled by configuring PERI_CRG26 bit[4].

3.3.3.4 Configuring the I²C Clock Frequency

[Table 3-6](#) describes frequency configurations of I²C clocks.

Table 3-6 Frequency configurations of I²C clocks

Signal	Description
i2c2_cken	I ² C2 clock status. This signal can be controlled by configuring PERI_CRG27 bit[12].
i2c1_cken	I ² C1 clock status. This signal can be controlled by configuring PERI_CRG27 bit[8].
i2c0_cken	I ² C0 clock status. This signal can be controlled by configuring PERI_CRG27 bit[4].

3.3.3.5 Configuring the SSP Clock Frequency

[Table 3-7](#) describes the frequency configuration of the SSP clock.

Table 3-7 Frequency configuration of the SSP clock

Signal	Description
ssp0_cken	SSP0 clock status. This signal can be controlled by configuring PERI_CRG28 bit[0].

3.3.3.6 Configuring the SCI Clock Frequency

[Table 3-8](#) describes frequency configuration of the SCI clock.

Table 3-8 Frequency configuration of the SCI clock

Signal	Description
sci0_cken	SCI0 clock state. This signal can be controlled by configuring PERI_CRG29 bit[0].



3.3.3.7 Configuring the VDH Clock Frequency

[Table 3-9](#) describes frequency configurations of VDH clocks.

Table 3-9 Frequency configurations of VDH clocks

Signal	Description
vdh_clk_sel	VDH clock select. This signal can be controlled by configuring PERI_CRG30 bit[9:8].
vdh_cken	VDH clock status. This signal can be controlled by configuring PERI_CRG30 bit[0].

3.3.3.8 Configuring the JPGD Clock Frequency

[Table 3-10](#) describes frequency configurations of JPGD clocks.

Table 3-10 Frequency configurations of JPGD clocks

Signal	Description
jpgd_clk_sel	JPGD clock select. This signal can be controlled by configuring PERI_CRG31 bit[8].
jpgd_cken	JPGD clock status. This signal can be controlled by configuring PERI_CRG31 bit[0].

3.3.3.9 Configuring the PGD Clock Frequency

[Table 3-11](#) describes the frequency configuration of the PGD clock.

Table 3-11 Frequency configuration of the PGD clock

Signal	Description
pgd_cken	PGD clock status. This signal can be controlled by configuring PERI_CRG33 bit[0].

3.3.3.10 Configuring the BPD Clock Frequency

[Table 3-12](#) describes the frequency configuration of the BPD clock.

Table 3-12 Frequency configuration of the BPD clock

Signal	Description
bpd_cken	BPD clock status. This signal can be controlled by configuring PERI_CRG34 bit[0].



3.3.3.11 Configuring the VENC Clock Frequency

Table 3-13 describes frequency configurations of VENC clocks.

Table 3-13 Frequency configurations of VENC clocks

Signal	Description
venc_clk_sel	VENC clock frequency select. This signal can be controlled by configuring PERI_CRG35 bit[8:7].
venc_cken	VENC clock status. This signal can be controlled by configuring PERI_CRG35 bit[0].

3.3.3.12 Configuring the JPGE Clock Frequency

Table 3-14 describes the frequency configuration of the JPGE clock.

Table 3-14 Frequency configuration of the JPGE clock

Signal	Description
jpge_cken	JPGE clock status. This signal can be controlled by configuring PERI_CRG36 bit[0].

3.3.3.13 Configuring the TDE Clock Frequency

Table 3-15 describes frequency configurations of TDE clocks.

Table 3-15 Frequency configurations of TDE clocks

Signal	Description
tde_clk_sel	TDE clock select. This signal can be controlled by configuring PERI_CRG37 bit[9:8].
tde_cken	TDE clock status. This signal can be controlled by configuring PERI_CRG37 bit[0].

3.3.3.14 Configuring the SDIO0 Clock Frequency

Table 3-16 describes frequency configurations of SDIO0 clocks.

Table 3-16 Frequency configurations of SDIO0 clocks

Signal	Description
sdio0_clk_sel	SDIO0 clock select. This signal can be controlled by configuring PERI_CRG39 bit[9:8].



Signal	Description
sdio0_cken	SDIO0 clock status. This signal can be controlled by configuring PERI_CRG39 bit[1].
sdio0_bus_cken	SDIO0 bus clock status. This signal can be controlled by configuring PERI_CRG39 bit[0].

3.3.3.15 Configuring the SDIO1 Clock Frequency

[Table 3-17](#) describes frequency configurations of SDIO1 clocks.

Table 3-17 Frequency configurations of SDIO1 clocks

Signal	Description
sdio1_clk_sel	SDIO1 clock select. This signal can be controlled by configuring PERI_CRG40 bit[9:8].
sdio1_cken	SDIO1 clock status. This signal can be controlled by configuring PERI_CRG40 bit[1].
sdio1_bus_cken	SDIO1 bus clock status. This signal can be controlled by configuring PERI_CRG40 bit[0].

3.3.3.16 Configuring the DMA Clock Frequency

[Table 3-18](#) describes the frequency configuration of the DMA clock.

Table 3-18 Frequency configuration of the DMA clock

Signal	Description
dma_cken	DMA clock status. This signal can be controlled by configuring PERI_CRG41 bit[0].

3.3.3.17 Configuring the USB3CTRL Clock Frequency

[Table 3-19](#) describes frequency configurations of USB3CTRL clocks.

Table 3-19 Frequency configurations of USB3CTRL clocks

Signal	Description
usb3_pcs_ref_cken	USB3CTRL PCS_REF clock status. This signal can be controlled by configuring PERI_CRG44 bit[9].
usb3_pcs_rx_cken	USB3CTRL PCS_RX clock status. This signal can be controlled by configuring PERI_CRG44 bit[8].



Signal	Description
usb3_pcs_pclk_cken	USB3CTRL PCS_PCLK clock status. This signal can be controlled by configuring PERI_CRG44 bit[7].
usb3_bus_gm_cken	USB3CTRL GM bus clock status. This signal can be controlled by configuring PERI_CRG44 bit[6].
usb3_bus_gs_cken	USB3CTRLGS bus clock status. This signal can be controlled by configuring PERI_CRG44 bit[5].
usb3_utmi_cken	USB3CTRL UTMI clock status. This signal can be controlled by configuring PERI_CRG44 bit[4].
usb3_suspend_cken	USB3CTRL SUSPEND clock status. This signal can be controlled by configuring PERI_CRG44 bit[2].
usb3_ref_cken	USB3CTRL REF clock status. This signal can be controlled by configuring PERI_CRG44 bit[1].
usb3_bus_cken	USB3CTRL bus clock status. This signal can be controlled by configuring PERI_CRG44 bit[0].

3.3.3.18 Configuring the USB2CTRL0 Clock Frequency

[Table 3-20](#) describes frequency configurations of USB2 CTRL0 clocks.

Table 3-20 Frequency configurations of USB2 CTRL0 clocks

Signal	Description
usb2_utmi0_cken	USB2 CTRL0 UTMIO clock status. This signal can be controlled by configuring PERI_CRG46 bit[5].
usb2_hst_phy_cken	USB2 CTRL0 HSTPHY clock status. This signal can be controlled by configuring PERI_CRG46 bit[4].
usb2_otg_phy_cken	USB2 CTRL0 OTGPHY clock status. This signal can be controlled by configuring PERI_CRG46 bit[3].
usb2_ohci12m_cken	USB2 CTRL0 OHCI12M clock status. This signal can be controlled by configuring PERI_CRG46 bit[2].
usb2_ohci48m_cken	USB2 CTRL0 OHCI48M clock status. This signal can be controlled by configuring PERI_CRG46 bit[1].
usb2_bus_cken	USB2 CTRL0 bus clock status. This signal can be controlled by configuring PERI_CRG46 bit[0].

3.3.3.19 Configuring the USB2 PHY0 Clock Frequency

[Table 3-21](#) describes frequency configurations of USB2 PHY0 clocks.



Table 3-21 Frequency configurations of USB2 PHY0 clocks

Signal	Description
usb2_phy_refclk_sel	USB2 PHY0 REFCLK select. This signal can be controlled by configuring PERI_CRG47 bit[16].
usb2_phy_ref_cken	USB2 PHY0 REFCLK status. This signal can be controlled by configuring PERI_CRG47 bit[0].

3.3.3.20 Configuring the CA Clock Frequency

[Table 3-22](#) describes the frequency configuration of the CA clock.

Table 3-22 Frequency configuration of the CA clock

Signal	Description
ca_ci_clk_sel	CA clock select. This signal can be controlled by configuring PERI_CRG48 bit[12].

3.3.3.21 Configuring the SHA Clock Frequency

[Table 3-23](#) describes the frequency configuration of the SHA clock.

Table 3-23 Frequency configuration of the SHA clock

Signal	Description
sha_cken	SHA clock status. This signal can be controlled by configuring PERI_CRG49 bit[0].

3.3.3.22 Configuring the PMC Clock Frequency

[Table 3-24](#) describes the frequency configuration of the PMC clock.

Table 3-24 Frequency configuration of the PMC clock

Signal	Description
pmc_cken	PMC clock status. This signal can be controlled by configuring PERI_CRG50 bit[0].

3.3.3.23 Configuring the ETH Clock Frequency

[Table 3-25](#) describes frequency configurations of ETH clocks.



Table 3-25 Frequency configurations of ETH clocks

Signal	Description
eth_clk_sel	ETH clock select. This signal can be controlled by configuring PERI_CRG51 bit[3:2].
eth_cken	ETH working clock status. This signal can be controlled by configuring PERI_CRG51 bit[1].
eth_bus_cken	ETH bus clock status. This signal can be controlled by configuring PERI_CRG51 bit[0].

3.3.3.24 Configuring the GPU Clock Frequency

[Table 3-26](#) describes frequency configurations of GPU clocks.

Table 3-26 Frequency configurations of GPU clocks

Signal	Description
gpu_cken	GPU clock status. This signal can be controlled by configuring PERI_CRG53 bit[0].
gpu_pp0_cken	GPU PP0 clock status. This signal can be controlled by configuring PERI_CRG53 bit[8].
gpu_pp1_cken	GPU PP1 clock status. This signal can be controlled by configuring PERI_CRG53 bit[9].

3.3.3.25 Configuring the VDP Clock Frequency

[Table 3-27](#) describes frequency configurations of VDP clocks.

Table 3-27 Frequency configurations of VDP clocks

Signal	Description
vo_hd_hdmi_clk_sel	Ratio of the HDMI clock frequency to the VO clock frequency of the HD channel. This signal can be controlled by configuring PERI_CRG54 bit[29].
vdp_clk_sel	VDP clock select. This signal can be controlled by configuring PERI_CRG54 bit[28].
vo_sd_hdmi_clk_sel	Ratio of the HDMI clock frequency to the VO clock frequency of the SD channel. This signal can be controlled by configuring PERI_CRG54 bit[27].
hdmi_clk_sel	HDMI clock select. This signal can be controlled by configuring PERI_CRG54 bit[26].
vdac_ch0_clk_sel	VDAC CH0 clock select. This signal can be controlled by configuring PERI_CRG54 bit[20].



Signal	Description
vo_hd_div	Ratio of the VO HD clock frequency to the VOHD_DATE clock frequency. This signal can be controlled by configuring PERI_CRG54 bit[19:18].
vo_hd_clk_sel	VOHD clock select. This signal can be controlled by configuring PERI_CRG54 bit[17:16].
vo_sd_clk_div	VO SD clock frequency divider select. This signal can be controlled by configuring PERI_CRG54 bit[15:14].
vo_sd_clk_sel	VOSD clock select. This signal can be controlled by configuring PERI_CRG54 bit[13:12].
vo_ch0_cken	VDAC CH0 clock status. This signal can be controlled by configuring PERI_CRG54 bit[6].
vo_hdate_cken	VO HD DATE clock status. This signal can be controlled by configuring PERI_CRG54 bit[5].
vo_hd_cken	VO HD clock status. This signal can be controlled by configuring PERI_CRG54 bit[4].
vo_sdate_cken	VO SD DATE clock status. This signal can be controlled by configuring PERI_CRG54 bit[3].
vo_sd_cken	VO SD clock status. This signal can be controlled by configuring PERI_CRG54 bit[2].
vo_cken	VO clock status. This signal can be controlled by configuring PERI_CRG54 bit[1].
vo_bus_cken	VO bus clock status. This signal can be controlled by configuring PERI_CRG54 bit[0].

3.3.3.26 Configuring the VPSS Clock Frequency

Table 3-28 describes frequency configurations of VPSS clocks.

Table 3-28 Frequency configurations of VPSS clocks

Signal	Description
vpss_clk_sel	VPSS clock select. This signal can be controlled by configuring PERI_CRG60 bit[9:8].
vpss_cken	VPSS clock status. This signal can be controlled by configuring PERI_CRG60 bit[0].

3.3.3.27 Configuring the PVR-1 Clock Frequency

Table 3-29 describes frequency configurations of PVR-1 clocks.



Table 3-29 Frequency configurations of PVR-1 clocks

Signal	Description
pvr_tsi2_pctrl	PVR TSI2 clock phase select. This signal can be controlled by configuring PERI_CRG63 bit[16].
pvr_tsi1_pctrl	PVR TSI1 clock phase select. This signal can be controlled by configuring PERI_CRG63 bit[15].
pvr_tsi2_cken	PVR TSI2 clock status. This signal can be controlled by configuring PERI_CRG63 bit[4].
pvr_tsi1_cken	PVR TSI1 clock status. This signal can be controlled by configuring PERI_CRG63 bit[3].
pvr_27m_cken	PVR 27 MHz clock status. This signal can be controlled by configuring PERI_CRG63 bit[2].
pvr_dmx_cken	PVR DMX clock status. This signal can be controlled by configuring PERI_CRG63 bit[1].
pvr_bus_cken	PVR bus clock status. This signal can be controlled by configuring PERI_CRG63 bit[0].

3.3.3.28 Configuring the PVR-2 Clock Frequency

[Table 3-30](#) describes the frequency configuration of the PVR-2 clock.

Table 3-30 Frequency configuration of the PVR-2 clock

Signal	Description
pvr_dmx_clk_sel	PVR DMX clock select. This signal can be controlled by configuring PERI_CRG64 bit[3:2].

3.3.3.29 Configuring the HDMITX CTRL Clock Frequency

[Table 3-31](#) describes frequency configurations of HDMI TX CTRL clocks.

Table 3-31 Frequency configurations of HDMI TX CTRL clocks

Signal	Description
hdmitx_ctrl_as_cken	HDMITX CTRL AS clock status. This signal can be controlled by configuring PERI_CRG67 bit[5].
hdmitx_ctrl_os_cken	HDMITX CTRL OS clock status. This signal can be controlled by configuring PERI_CRG67 bit[4].
hdmitx_ctrl_mhl_cken	HDMITX CTRL MHL clock status. This signal can be controlled by configuring PERI_CRG67 bit[3].
hdmitx_ctrl_id_cken	HDMITX CTRL ID clock status. This signal can be controlled by configuring PERI_CRG67 bit[2].



Signal	Description
hdmitx_ctrl_cec_cken	HDMITX CTRL CEC clock status. This signal can be controlled by configuring PERI_CRG67 bit[1].
hdmitx_ctrl_bus_cken	HDMITX CTRL bus clock status. This signal can be controlled by configuring PERI_CRG67 bit[0].

3.3.3.30 Configuring the HDMITX PHY Clock Frequency

[Table 3-32](#) describes the frequency configuration of the HDMI TX PHY clock.

Table 3-32 Frequency configuration of the HDMI TX PHY clock

Signal	Description
hdmitx_phy_bus_cken	HDMITX PHY bus clock status. This signal can be controlled by configuring PERI_CRG68 bit[0].

3.3.3.31 Configuring the ADAC Clock Frequency

[Table 3-33](#) describes the frequency configuration of the ADAC clock.

Table 3-33 Frequency configuration of the ADAC clock

Signal	Description
adac_cken	ADAC bus clock status. This signal can be controlled by configuring PERI_CRG69 bit[0].

3.3.3.32 Configuring the AIAO Clock Frequency

[Table 3-34](#) describes frequency configurations of AIAO clocks.

Table 3-34 Frequency configurations of AIAO clocks

Signal	Description
aiao_mclk_sel	MCLK clock source select. This signal can be controlled by configuring PERI_CRG70 bit[21:20].
aiao_cken	AIAO clock status. This signal can be controlled by configuring PERI_CRG70 bit[0].

3.3.3.33 Configuring the VDAC Clock Frequency

[Table 3-35](#) describes frequency configurations of VDAC clocks.



Table 3-35 Frequency configurations of the VDAC clocks

Signal	Description
vdac_c_clk_pctrl	VDAC C clock phase select. This signal can be controlled by configuring PERI_CRG71 bit[16].
vdac_chop_cken	VDAC CHOP clock status. This signal can be controlled by configuring PERI_CRG71 bit[0].

3.3.3.34 Configuring the FE PHY Clock Frequency

[Table 3-36](#) describes frequency configurations of FE PHY clocks.

Table 3-36 Frequency configurations of FE PHY clocks

Signal	Description
fephys_clk_sel	FE PHY clock select. This signal can be controlled by configuring PERI_CRG72 bit[8].
fephys_cken	FE PHY clock status. This signal can be controlled by configuring PERI_CRG72 bit[0].

3.3.3.35 Configuring the GPU Clock Frequency

[Table 3-37](#) describes the frequency configuration of the GPU clock.

Table 3-37 Frequency configuration of the GPU clock

Signal	Description
gpu_freq_sel_cfg_crg	GPU clock source select. This signal can be controlled by configuring PERI_CRG73 bit[2:0].

3.3.3.36 Configuring the DDR Clock Frequency

[Table 3-38](#) describes the frequency configuration of the DDR clock.

Table 3-38 Frequency configuration of the DDR clock

Signal	Description
ddr_freq_sel_cfg_crg	DDR clock source select. This signal can be controlled by configuring PERI_CRG74 bit[2:0].

3.3.3.37 Configuring the WDG Clock Frequency

[Table 3-39](#) describes frequency configurations of WDG clocks.



Table 3-39 Frequency configurations of WDG clocks

Signal	Description
wdg0_cken	WDG clock status. This signal can be controlled by configuring PERI_CRG94 bit[0].

3.3.3.38 Configuring the USB2 PHY1 Clock Frequency

[Table 3-40](#) describes frequency configurations of USB2 PHY1 clocks.

Table 3-40 Frequency configurations of USB2 PHY1 clocks

Signal	Description
usb2_phy1_refclk_sel	USB2 PHY1 REFCLK clock select. This signal can be controlled by configuring PERI_CRG100 bit[16].
usb2_phy1_ref_cken	USB2 PHY1 REFCLK clock status. This signal can be controlled by configuring PERI_CRG100 bit[0].

3.3.3.39 Configuring the USB3 PHY Clock Frequency

[Table 3-41](#) describes frequency configurations of USB3 PHY clocks.

Table 3-41 Frequency configurations of USB3 PHY clocks

Signal	Description
usb3_ref_ssp_en	USB3 PHY super-speed clock status. This signal can be controlled by configuring PERI_CRG101 bit[10].
usb3_ref_use_pad	USB3 PHY reference clock select. This signal can be controlled by configuring PERI_CRG101 bit[9].
usb3_phy_refclk_sel	USB3 PHY reference clock select. This signal can be controlled by configuring PERI_CRG101 bit[8].
usb3_phy_ref_cken	USB3 PHY reference clock status. This signal can be controlled by configuring PERI_CRG101 bit[0].

3.3.3.40 Configuring the USB2 CRTL1 Clock Frequency

[Table 3-42](#) describes frequency configurations of USB2 CTRL1 clocks.

Table 3-42 Frequency configurations of USB2 CTRL1 clocks

Signal	Description
usb2_utmi0_cken1	USB2 CTRL1 UTMIO clock status. This signal can be controlled by configuring PERI_CRG102 bit[5].



Signal	Description
usb2_hst_phy_cken1	USB2 CTRL1 HSTPHY clock status. This signal can be controlled by configuring PERI_CRG102 bit[4].
usb2_ohci12m_cken1	USB2 CTRL1 OHCI12M clock status. This signal can be controlled by configuring PERI_CRG102 bit[2].
usb2_ohci48m_cken1	USB2 CTRL1 OHCI48M clock status. This signal can be controlled by configuring PERI_CRG102 bit[1].
usb2_bus_cken1	USB2 CTRL1 bus clock status. This signal can be controlled by configuring PERI_CRG102 bit[0].

3.3.4 Register Summary

[Table 3-43](#) describes CRG registers.

Table 3-43 Summary of CRG registers (base address: 0xF8A2_2000)

Offset Address	Register	Description	Page
0x0058	PERI_CRG22	SYS clock and soft reset control register	3-25
0x0060	PERI_CRG24	NANDC clock and soft reset control register	3-26
0x0064	PERI_CRG25	DDR clock and soft reset control register	3-27
0x0068	PERI_CRG26	UART clock and soft reset control register	3-28
0x006C	PERI_CRG27	I ² C clock and soft reset control register	3-29
0x0070	PERI_CRG28	SSP clock and soft reset control register	3-30
0x0074	PERI_CRG29	SCI clock and soft reset control register	3-30
0x0078	PERI_CRG30	VDH clock and soft reset control register	3-31
0x007C	PERI_CRG31	JPGD clock and soft reset control register	3-32
0x0084	PERI_CRG33	PGD clock and soft reset control register	3-33
0x0088	PERI_CRG34	BPD clock and soft reset control register	3-33
0x008C	PERI_CRG35	VENC clock and soft reset control register	3-34
0x0090	PERI_CRG36	JPGE clock and soft reset control register	3-35
0x0094	PERI_CRG37	TDE clock and soft reset control register	3-36
0x009C	PERI_CRG39	SDIO0 (AHB1) clock and soft reset control register	3-37
0x00A0	PERI_CRG40	SDIO1 (AHB0) clock and soft reset control register	3-38



Offset Address	Register	Description	Page
0x00A4	PERI_CRG41	DMA clock and soft reset control register	3-40
0x00B0	PERI_CRG44	USB3 CTRL clock and soft reset control register	3-41
0x00B8	PERI_CRG46	USB2 CTRL0 clock and soft reset control register	3-42
0x00BC	PERI_CRG47	USB2 PHY0 clock and soft reset control register	3-44
0x00C0	PERI_CRG48	CA clock and soft reset control register	3-45
0x00C4	PERI_CRG49	SHA clock and soft reset control register	3-45
0x00C8	PERI_CRG50	PMC clock and soft reset control register	3-46
0x00CC	PERI_CRG51	ETH clock and soft reset control register	3-47
0x00D4	PERI_CRG53	GPU clock and soft reset control register	3-48
0x00D8	PERI_CRG54	VDP clock and soft reset control register	3-48
0x00F0	PERI_CRG60	VPSS clock and soft reset control register.	3-50
0x00FC	PERI_CRG63	PVR-1 clock and soft reset control register	3-52
0x0100	PERI_CRG64	PVR-2 clock and soft reset control register	3-53
0x010C	PERI_CRG67	HDMITX_CTRL clock and soft reset control register	3-54
0x0110	PERI_CRG68	HDMITX_PHY clock and soft reset control register	3-55
0x0114	PERI_CRG69	ADAC clock and soft reset control register	3-56
0x0118	PERI_CRG70	AIAO clock and soft reset control register	3-56
0x011C	PERI_CRG71	VDAC clock and soft reset control register	3-57
0x0120	PERI_CRG72	FE PHY clock and soft reset control register	3-58
0x0124	PERI_CRG73	GPU_LP clock and soft reset control register	3-59
0x0128	PERI_CRG74	DDR_LP clock and soft reset control register	3-60
0x0154	PERI_CRG85	SW_READBACK register	3-61
0x0168	PERI_CRG90	Output reset control register	3-62
0x016C	PERI_CRG91	USB_FREECLK_DEC control register	3-63
0x0174	PERI_CRG93	VDH_RST_READBACK register	3-63



Offset Address	Register	Description	Page
0x0178	PERI_CRG94	WDG clock and soft reset control register	3-64
0x0190	PERI_CRG100	USB2 PHY2 clock and soft reset control register	3-66
0x0194	PERI_CRG101	USB3 PHY clock and soft reset control register	3-67
0x0198	PERI_CRG102	USB2 CTRL2 clock and soft reset control register	3-68

3.3.5 Register Description

PERI_CRG22

PERI_CRG22 is a SYS clock and soft reset control register.

	Offset Address	Register Name	Total Reset Value																
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	PERI_CRG22	0x0000_0000																
Name	reserved												core_bus_clk_div	mde3_clk_sel	mde2_clk_sel	reserved	mde1_clk_sel	mde0_clk_sel	core_bus_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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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]	RW	reserved	Reserved																
[13:12]	RW	core_bus_clk_div	Core bus clock frequency divider 00: divided by 1 01: divided by 2 10: divided by 3 11: divided by 4																
[11:10]	RW	mde3_clk_sel	Media3 bus clock select 00: 24 MHz 01: 300 MHz 10: 400 MHz 11: 345.6 MHz																



[9:8]	RW	mde2_clk_sel	Media2 bus clock select 00: 24 MHz 01: 300 MHz 10: 400 MHz 11: 345.6 MHz
[7:6]	RW	reserved	Reserved
[5:4]	RW	mde1_clk_sel	Media1 bus clock select 00: 24 MHz 01: 200 MHz 10: 250 MHz 11: reserved
[3:2]	RW	mde0_clk_sel	Media0 bus clock select 00: 24 MHz 01: 300 MHz 10: 400 MHz 11: 345.6 MHz
[1:0]	RW	core_bus_clk_sel	Core bus clock select 00: 24 MHz 01: 200 MHz 10: 100 MHz 11: reserved

PERI_CRG24

PERI_CRG24 is a NANDC clock and soft reset control register.

Offset Address		Register Name		Total Reset Value				
Bit	31 30 29 28	27 26 25 24	23 22 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_clk_sel	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					
[31:11]	RW	reserved	Reserved					



[10:8]	RW	nf_clk_sel	NANDC clock select 000: 24 MHz 100: 200 MHz Others: reserved
[7:5]	RW	reserved	Reserved
[4]	RW	nf_srst_req	NANDC soft reset 0: not reset 1: reset
[3:1]	RW	reserved	Reserved
[0]	RW	nf_cken	NANDC clock gating 0: disabled 1: enabled

PERI_CRG25

PERI_CRG25 is a DDR clock and soft reset control register.

	Offset Address 0x0064																Register Name PERI_CRG25								Total Reset Value 0x0000_100F										
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	reserved																									ddrt_cken	reserved	ddrt_srst_req	ddrc_srst_req	hipack_srst_req	reserved	ddrc_cken	hipack_cken	ddphy_cken	drr_cken
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	1	1	1	1			
Bits	Access	Name			Description																														
[31:13]	RW	reserved			Reserved																														
[12]	RW	ddrt_cken			DDRT clock gating 0: disabled 1: enabled																														
[11]	RW	reserved			Reserved																														
[10]	RW	ddrt_srst_req			DDRT soft reset 0: not reset 1: reset																														



[9]	RW	ddrc_srst_req	DDRC prst soft reset 0: not reset 1: reset
[8]	RW	hipack_srst_req	HIPACK prst soft reset 0: not reset 1: reset
[7:4]	RW	reserved	Reserved
[3]	RW	ddrc_cken	DDRC pclk clock gating 0: disabled 1: enabled
[2]	RW	hipack_cken	HIPACK pclk clock gating 0: disabled 1: enabled
[1]	RW	ddrphy_cken	DDR PHY clock gating 0: disabled 1: enabled
[0]	RW	ddr_cken	DDRC clock gating 0: disabled 1: enabled

PERI_CRG26

PERI_CRG26 is a UART clock and soft reset control register.

	Offset Address 0x0068																Register Name PERI_CRG26								Total Reset Value 0x0000_0011							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									uart2_srst_req	uart2_cken	reserved	uart1_srst_req	uart1_cken		
Reset	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	1			
Bits	Access	Name			Description																											
[31:6]	RW	reserved			Reserved																											



[5]	RW	uart2_srst_req	UART2 soft reset 0: not reset 1: reset
[4]	RW	uart2_cken	UART2 clock gating 0: disabled 1: enabled
[3:2]	RW	reserved	Reserved
[1]	RW	uart1_srst_req	UART1 soft reset 0: not reset 1: reset
[0]	RW	uart1_cken	UART1 clock gating 0: disabled 1: enabled

PERI_CRG27

PERI_CRG27 is an I²C clock and soft reset control register.

	Offset Address 0x006C																Register Name PERI_CRG27								Total Reset Value 0x0000_1110							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																i2c2_srst_req	i2c2_cken	reserved	i2c1_srst_req	i2c1_cken	reserved	i2c0_srst_req	i2c0_cken	reserved							
Reset	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	0	
Bits	Access	Name		Description																												
[31:14]	RW	reserved		Reserved																												
[13]	RW	i2c2_srst_req		I ² C2 soft reset 0: not reset 1: reset																												
[12]	RW	i2c2_cken		I ² C2 clock gating 0: disabled 1: enabled																												
[11:10]	RW	reserved		Reserved																												



[9]	RW	i2c1_srst_req	I ² C1 soft reset 0: not reset 1: reset
[8]	RW	i2c1_cken	I ² C1 clock gating 0: disabled 1: enabled
[7:6]	RW	reserved	Reserved
[5]	RW	i2c0_srst_req	I ² C0 soft reset 0: not reset 1: reset
[4]	RW	i2c0_cken	I ² C0 clock gating 0: disabled 1: enabled
[3:0]	RW	reserved	Reserved

PERI_CRG28

PERI_CRG28 is an SSP clock and soft reset control register.

	Offset Address 0x0070																Register Name PERI_CRG28								Total Reset Value 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																												ssp0_srst_req	ssp0_cken		
Reset	0	0	0	0	0	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]	RW	reserved			Reserved																											
[1]	RW	ssp0_srst_req			SSP0 soft reset 0: not reset 1: reset																											
[0]	RW	ssp0_cken			SSP0 clock gating 0: disabled 1: enabled																											



PERI_CRG29

PERI_CRG29 is an SCI clock and soft reset control register.

	Offset Address	Register Name	Total Reset Value															
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	PERI_CRG29	0x0000_0001															
Name	reserved																sci0_srst_req	sci0_cken
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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]	RW	reserved	Reserved															
[1]	RW	sci0_srst_req	SCI0 soft rest 0: not reset 1: reset															
[0]	RW	sci0_cken	SCI0 clock gating 0: disabled 1: enabled															

PERI_CRG30

PERI_CRG30 is a VDH clock and soft reset control register.



Offset Address												Register Name												Total Reset Value											
0x0078												PERI_CRG30												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												vdhclk_loaden	vdhclk_skipcfg				reserved	vdh_clk_sel	reserved	vdh_mfd_srst_req	vdh_scd_srst_req	vdh_all_srst_req	reserved	vdhdsp_cken	vdh_cken									
Reset	0	0	0	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																																
[31:18]	RW	reserved	Reserved																																
[17]	RW	vdhclk_loaden	VDH pulse filter/frequency scale enable 0: disabled 1: enabled																																
[16:12]	RW	vdhclk_skipcfg	VDH pulse filter/frequency scale 0x0: No pulse is filtered out. 0x1: One pulse is filtered out. 0x2: Two pulses are filtered out. The rule applies to other values.																																
[11:10]	RW	reserved	Reserved																																
[9:8]	RW	vdh_clk_sel	VDH clock select 00: 300 MHz 01: 288 MHz 10: 400 MHz 11: 345.6 MHz																																
[7]	RW	reserved	Reserved																																
[6]	RW	vdh_mfd_srst_req	MFD soft reset 0: not reset 1: reset																																
[5]	RW	vdh_scd_srst_req	SCD soft reset 0: not reset 1: reset																																
[4]	RW	vdh_all_srst_req	VDH soft reset 0: not reset 1: reset																																
[3:2]	RW	reserved	Reserved																																



[1]	RW	vdhdsp_cken	VDHDSP clock gating 0: disabled 1: enabled
[0]	RW	vdh_cken	VDH clock gating 0: disabled 1: enabled

PERI_CRG31

PERI_CRG31 is a JPGD clock and soft reset control register.

	Offset Address 0x007C																Register Name PERI_CRG31								Total Reset Value 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																									jpgd_clk_sel	reserved	jpgd_srst_req	reserved	jpgd_cken		
Reset	0	0	0	0	0	0	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																											
[31:9]	RW	reserved			Reserved																											
[8]	RW	jpgd_clk_sel			JPGD clock select 0: 200 MHz 1: 150 MHz																											
[7:5]	RW	reserved			Reserved																											
[4]	RW	jpgd_srst_req			JPGD soft reset 0: not reset 1: reset																											
[3:1]	RW	reserved			Reserved																											
[0]	RW	jpgd_cken			JPGD clock gating 0: disabled 1: enabled																											

PERI_CRG33

PERI_CRG33 is a PGD clock and soft reset control register.



PERI_CRG34

PERI_CRG34 is a BPD clock and soft reset control register.

Offset Address								Register Name								Total Reset Value																
0x0088								PERI_CRG34								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																									bpd_srst_req	reserved	bpd_cken				
Reset	0	0	0	0	0	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]	RW	reserved				Reserved																										



[4]	RW	bpd_srst_req	BPD soft reset 0: not reset 1: reset
[3:1]	RW	reserved	Reserved
[0]	RW	bpd_cken	BPD clock gating 0: disabled 1: enabled

PERI_CRG35

PERI_CRG35 is a VENC clock and soft reset control register.

	Offset Address 0x008C																Register Name PERI_CRG35								Total Reset Value 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																								venc_clk_sel	reserved	venc_srst_req	reserved	venc_cken			
Reset	0	0	0	0	0	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																												
[31:9]	RW	reserved		Reserved																												
[8:7]	RW	venc_clk_sel		VENC clock select 00: 200 MHz 10: 150 MHz X1: reserved																												
[6:5]	RW	reserved		Reserved																												
[4]	RW	venc_srst_req		VENC soft reset 0: not reset 1: reset																												
[3:1]	RW	reserved		Reserved																												
[0]	RW	venc_cken		VENC clock gating 0: disabled 1: enabled																												



PERI_CRG36

PERI_CRG36 is a JPGE clock and soft reset control register.

	Offset Address	Register Name	Total Reset Value												
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	PERI_CRG36	0x0000_0001												
Name	reserved												jpge_srst_req	reserved	jpge_cken
Reset	0 0 0 0 0 0 0 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]	RW	reserved	Reserved												
[4]	RW	jpge_srst_req	JPGE soft reset 0: not reset 1: reset												
[3:1]	RW	reserved	Reserved												
[0]	RW	jpge_cken	JPGE clock gating 0: disabled 1: enabled												

PERI_CRG37

PERI_CRG37 is a TDE clock and soft reset control register.



Offset Address												Register Name												Total Reset Value											
0x0094												PERI_CRG37												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												tdeclk_loaden	tdeclk_skipcfg				reserved	tde_clk_sel	reserved	tde_srst_req	reserved	tde_cken	reserved	reserved	reserved	reserved	reserved	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	1					
Bits	Access	Name			Description																														
[31:18]	RW	reserved			Reserved																														
[17]	RW	tdeclk_loaden			TDE pulse filter/frequency scale enable 0: disabled 1: enabled																														
[16:12]	RW	tdeclk_skipcfg			TDE pulse filter/frequency scale 0x0: No pulse is filtered out. 0x1: One pulse is filtered out. 0x2: Two pulses are filtered out. The rule applies to other values.																														
[11:10]	RW	reserved			Reserved																														
[9:8]	RW	tde_clk_sel			TDE clock select 00: 400 MHz 01: 500 MHz 1X: 432 MHz																														
[7:5]	RW	reserved			Reserved																														
[4]	RW	tde_srst_req			TDE soft reset 0: not reset 1: reset																														
[3:1]	RW	reserved			Reserved																														
[0]	RW	tde_cken			TDE clock gating 0: disabled 1: enabled																														

PERI_CRG39

PERI_CRG39 is an SDIO0 (AHB1) clock and soft reset control register.



Offset Address												Register Name												Total Reset Value											
0x009C												PERI_CRG39												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												sdio0_clk_mode	sdio0_drv_ps_sel	reserved	sdio0_sap_ps_sel	reserved	sdio0_clk_sel	reserved	sdio0_srst_req	reserved	sdio0_cken	sdio0_bus_cken												
Reset	0	0	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																															
[31:20]	RW	reserved		Reserved																															
[19]	RW	sdio0_clk_mode		SDIO0 phase select 0: normal mode 1: DDR50 mode (reserved)																															
[18:16]	RW	sdio0_drv_ps_sel		Phase shift of the SDIO0 driver clock 000: 0° 001: 45° 010: 90° 011: 135° 100: 180° 101: 225° 110: 270° 111: 315°																															
[15]	RW	reserved		Reserved																															
[14:12]	RW	sdio0_sap_ps_sel		Phase shift of the SDIO0 sampling clock 000: 0° 001: 45° 010: 90° 011: 135° 100: 180° 101: 225° 110: 270° 111: 315°																															
[11:10]	RW	reserved		Reserved																															



[9:8]	RW	sdio0_clk_sel	SDIO0 clock select 00: 75 MHz 01: 100 MHz 10: 50 MHz 11: 25 MHz
[7:5]	RW	reserved	Reserved
[4]	RW	sdio0_srst_req	SDIO0 soft reset 0: not reset 1: reset
[3:2]	RW	reserved	Reserved
[1]	RW	sdio0_cken	SDIO0 clock gating 0: disabled 1: enabled
[0]	RW	sdio0_bus_cken	SDIO0 bus clock gating 0: disabled 1: enabled

PERI_CRG40

PERI_CRG40 is an SDIO1 (AHB0) clock and soft reset control register.

	Offset Address 0x00A0										Register Name PERI_CRG40								Total Reset Value 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										sdio1_clk_mode	sdio1_drv_ps_sel	reserved	sdio1_sap_ps_sel	reserved	sdio1_clk_sel	reserved	sdio1_srst_req	reserved	sdio1_cken	sdio1_bus_cken											
Reset	0	0	0	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																											
[31:20]	RW	reserved			Reserved																											
[19]	RW	sdio1_clk_mode			SDIO1 phase select 0: normal mode 1: DDR50 mode																											



[18:16]	RW	sdio1_drv_ps_sel	Phase shift of the SDIO1 driver clock (normal mode) 000: 0° 001: 45° 010: 90° 011: 135° 100: 180° 101: 225° 110: 270° 111: 315° Phase shift of the SDIO1 driver clock (DDR50 mode) 000: 0° 001: 22.5° 010: 45° 011: 67.5° 100: 90° 101: 112.5° 110: 145° 111: 167.5°
[15]	RW	reserved	Reserved
[14:12]	RW	sdio1_sap_ps_sel	Phase shift of the SDIO1 sampling clock (normal mode) 000: 0° 001: 45° 010: 90° 011: 135° 100: 180° 101: 225° 110: 270° 111: 315° Phase shift of the SDIO1 sampling clock (DDR50 mode) 000: 0° 001: 22.5° 010: 45° 011: 67.5° 100: 90° 101: 112.5° 110: 145° 111: 167.5°
[11:10]	RW	reserved	Reserved



[9:8]	RW	sdio1_clk_sel	SDIO1 clock select 00: 75 MHz 01: 100 MHz 10: 50 MHz 11: 25 MHz
[7:5]	RW	reserved	Reserved
[4]	RW	sdio1_srst_req	SDIO1 soft reset 0: not reset 1: reset
[3:2]	RW	reserved	Reserved
[1]	RW	sdio1_cken	SDIO1 clock gating 0: disabled 1: enabled
[0]	RW	sdio1_bus_cken	SDIO1 bus clock gating 0: disabled 1: enabled

PERI_CRG41

PERI_CRG41 is a DMA clock and soft reset control register.

	Offset Address 0x00A4																Register Name PERI_CRG41								Total Reset Value 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																													dmac_srst_req	reserved	dmac_cken	
Reset	0	0	0	0	0	0	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]	RW	reserved				Reserved																											
[4]	RW	dmac_srst_req				DMA soft reset 0: not reset 1: reset																											
[3:1]	RW	reserved				Reserved																											



[0]	RW	dmac_cken	DMA clock gating 0: disabled 1: enabled
-----	----	-----------	---

PERI_CRG44

PERI_CRG44 is a USB3 CTRL clock and soft reset control register.

	Offset Address 0x00B0																Register Name PERI_CRG44								Total Reset Value 0x0000_13F7														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Name	reserved																																						
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	1	1	1	1	1	0	1	1	1					
Bits	Access	Name		Description																																			
[31:13]	RW	reserved		Reserved																																			
[12]	RW	usb3_vcc_srst_req		USB3 CTRL VCC soft reset 0: not reset 1: reset																																			
[11:10]	RW	reserved		Reserved																																			
[9]	RW	usb3_pcs_ref_cken		USB3 CTRL PCS_REF clock gating 0: disabled 1: enabled																																			
[8]	RW	usb3_pcs_rx_cken		USB3 CTRL PCS RX clock gating 0: disabled 1: enabled																																			
[7]	RW	usb3_pcs_pclk_cken		USB3 CTRL PCS PCLK clock gating 0: disabled 1: enabled																																			
[6]	RW	usb3_bus_gm_cken		USB3 CTRL GM bus clock gating 0: disabled 1: enabled																																			



[5]	RW	usb3_bus_gs_cken	USB3 CTRL GS bus clock gating 0: disabled 1: enabled
[4]	RW	usb3_utmi_cken	USB3 CTRL UTMI clock gating 0: disabled 1: enabled
[3]	RW	reserved	Reserved
[2]	RW	usb3_suspend_cken	USB3 CTRL suspend clock gating 0: disabled 1: enabled
[1]	RW	usb3_ref_cken	USB3 CTRL reference clock gating 0: disabled 1: enabled
[0]	RW	usb3_bus_cken	USB3 CTRL bus clock gating 0: disabled 1: enabled

PERI_CRG46

PERI_CRG46 is a USB2 CTRL0 clock and soft reset control register.

Offset Address 0x00B8												Register Name PERI_CRG46												Total Reset Value 0x0013_707F												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved												usb2_clk48_sel	reserved	usb2_otg_phy_srst_req	usb2_hst_phy_srst_req	reserved	usb2_utmi1_srst_req	usb2_utmi0_srst_req	usb2_bus_srst_req	reserved	usb2_utmi1_cken	usb2_utmi0_cken	usb2_hst_phy_cken	usb2_otg_utmi_cken	usb2_ohci12m_cken	usb2_ohci48m_cken	usb2_bus_cken								
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1					
Bits	Access	Name			Description																															
[31:21]	RW	reserved			Reserved																															
[20]	RW	usb2_clk48_sel			USB2 CLK48 clock source select 0: provided by the CRG 1: PHY output																															
[19:18]	RW	reserved			Reserved																															



[17]	RW	usb2_otg_phy_srst_req	USB2 CTRL OTG_PHY soft reset 0: not reset 1: reset
[16]	RW	usb2_hst_phy_srst_req	USB2 CTRL HST_PHY soft reset 0: not reset 1: reset
[15]	RW	reserved	Reserved
[14]	RW	usb2_utmi1_srst_req	USB2 CTRL UTMI1 soft reset 0: not reset 1: reset
[13]	RW	usb2_utmi0_srst_req	USB2 CTRL UTMI0 soft reset 0: not reset 1: reset
[12]	RW	usb2_bus_srst_req	USB2 CTRL bus soft reset 0: not reset 1: reset
[11:7]	RW	reserved	Reserved
[6]	RW	usb2_utmi1_cken	USB2 CTRL UTMI1 clock gating 0: disabled 1: enabled
[5]	RW	usb2_utmi0_cken	USB2 CTRL UTMI0 clock gating 0: disabled 1: enabled
[4]	RW	usb2_hst_phy_cken	USB2 CTRL HST_PHY clock gating 0: disabled 1: enabled
[3]	RW	usb2_otg_utmi_cken	USB2 CTRL OTG_UTMI clock gating 0: disabled 1: enabled
[2]	RW	usb2_ocih12m_cken	USB2 CTRL OHCI12M clock gating 0: disabled 1: enabled
[1]	RW	usb2_ocih48m_cken	USB2 CTRL OHCI48M clock gating 0: disabled 1: enabled



[0]	RW	usb2_bus_cken	USB2 CTRL bus clock gating 0: disabled 1: enabled
-----	----	---------------	---

PERI_CRG47

PERI_CRG47 is a USB2 PHY0 clock and soft reset control register.

Offset Address								Register Name								Total Reset Value																
0x00BC								PERI_CRG47								0x0000_0B01																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
Name	reserved																usb2_phy_refclk_sel		reserved		usb2_phy_srst_treq1		reserved		usb2_phy_srst_treq0		usb2_phy_srst_req		reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1
Bits	Access	Name		Description																												
[31:17]	RW	reserved		Reserved																												
[16]	RW	usb2_phy_refclk_sel		USB2 PHY reference clock select 0: crystal oscillator clock 1: 24 MHz internal clock																												
[15:12]	RW	reserved		Reserved																												
[11]	RW	usb2_phy_srst_treq1		USB2 PHY1 TPOR soft reset 0: not reset 1: reset																												
[10]	RW	reserved		Reserved																												
[9]	RW	usb2_phy_srst_treq0		USB2 PHY0 TPOR soft reset 0: not reset 1: reset																												
[8]	RW	usb2_phy_srst_req		USB2 PHY POR soft reset 0: not reset 1: reset																												
[7:1]	RW	reserved		Reserved																												



[0]	RW	usb2_phy_ref_cken	USB2 PHY reference clock gating 0: disabled 1: enabled
-----	----	-------------------	--

PERI_CRG48

PERI_CRG48 is a CA clock and soft reset control register.

	Offset Address 0x00C0																Register Name PERI_CRG48								Total Reset Value 0x0000_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved																ca_ci_clk_sel	reserved	otp_srst_req	ca_ci_srst_req	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	0			
Bits	Access	Name		Description																																
[31:13]	RW	reserved		Reserved																																
[12]	RW	ca_ci_clk_sel		CA cipher clock select 0: 250 MHz 1: 150 MHz																																
[11]	RW	reserved		Reserved																																
[10]	RW	otp_srst_req		OTP soft reset 0: not reset 1: reset																																
[9]	RW	ca_ci_srst_req		CA cipher soft reset 0: not reset 1: reset																																
[8:0]	RW	reserved		Reserved																																

PERI_CRG49

PERI_CRG49 is an SHA clock and soft reset control register.



PERI_CRG50

PERI_CRG50 is a PMC clock and soft reset control register.

Offset Address								Register Name								Total Reset Value																
0x00C8								PERI_CRG50								0x0000_0011																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									pmc_srst_req	reserved		pmc_cken			
Reset	0	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	1		
Bits	Access	Name			Description																											
[31:5]	RW	reserved			Reserved																											



[4]	RW	pmc_srst_req	PMC soft reset 0: not reset 1: reset
[3:1]	RW	reserved	Reserved
[0]	RW	pmc_cken	PMC clock gating 0: disabled 1: enabled

PERI_CRG51

PERI_CRG51 is an ETH clock and soft reset control register.

	Offset Address 0x00CC																Register Name PERI_CRG51								Total Reset Value 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																												hrst_eth_s	eth_clk_sel	eth_cken	eth_bus_cken			
Reset	0	0	0	0	0	0	0	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																															
[31:5]	RW	reserved		Reserved																															
[4]	RW	hrst_eth_s		ETH reset (only when the ETH bus is idle) 0: not reset 1: reset																															
[3:2]	RW	eth_clk_sel		ETH working clock select 0: 54 MHz 1: 27 MHz 2: 83.3 MHz																															
[1]	RW	eth_cken		ETH working clock gating 0: disabled 1: enabled																															
[0]	RW	eth_bus_cken		ETH bus clock gating 0: disabled 1: enabled																															



PERI_CRG53

PERI_CRG53 is a GPU clock and soft reset control register.

	Offset Address																Register Name								Total Reset Value							
	0x00D4																PERI_CRG53								0x0000_0711							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																													gpu_cken		
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	1	0	0	0	1
Bits	Access	Name	Description																													
[31:11]	RW	reserved	Reserved																													
[10]	RW	gpu_pp2_cken	PP1 clock gating 0: disabled 1: enabled																													
[9]	RW	gpu_pp1_cken	PP0 clock gating 0: disabled 1: enabled																													
[8]	RW	gpu_pp0_cken	GP clock gating 0: disabled 1: enabled																													
[7:5]	RW	reserved	Reserved																													
[4]	RW	gpu_srst_req	GPU module soft reset 0: not reset 1: reset																													
[3:1]	RW	reserved	Reserved																													
[0]	RW	gpu_cken	GPU clock gating 0: disabled 1: enabled																													

PERI_CRG54

PERI_CRG54 is a VDP clock and soft reset control register.



Offset Address 0x00D8												Register Name PERI_CRG54												Total Reset Value 0x0401_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	vou_srst_req	vo_hd_hdmi_clk_sel	vdp_clk_sel	vo_sd_hdmi_clk_sel	hdmi_clk_sel	reserved						vdac_ch0_clk_sel	vo_hd_clk_div	vo_hd_clk_sel	vo_sd_clk_div	vo_sd_clk_sel	reserved						vdac_ch0_cken	vo_hdiate_cken	vo_hd_cken	vo_sdiate_cken	vo_sd_cken	vo_cken	vo_bus_cken					
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1				
Bits	Access	Name		Description																															
[31]	RW	reserved		Reserved																															
[30]	RW	vou_srst_req		VOU soft reset 0: not reset 1: reset																															
[29]	RW	vo_hd_hdmi_clk_sel		Ratio of clk_vo_hd to clk_hdmi 0: 1:1 1: 1:2																															
[28]	RW	vdp_clk_sel		VDP clock select 0: 300 MHz 1: 200 MHz																															
[27]	RW	vo_sd_hdmi_clk_sel		Ratio of clk_vo_sd to clk_hdmi 0: 1:1 1: 1:2																															
[26]	RW	hdmi_clk_sel		HDMI clock select 0: clk_vo_sd 1: clk_vo_hd																															
[25:21]	RW	reserved		Reserved																															
[20]	RW	vdac_ch0_clk_sel		VDAC channel 0 clock select 0: clk_date_sd 1: clk_date_hd																															
[19:18]	RW	vo_hd_clk_div		VO HD clock divider 00: divided by 2 01: divided by 4 1X: divided by 1																															



[17:16]	RW	vo_hd_clk_sel	VO HD clock select 00: clk_vo_sd_ini 01: clk_vo_hd0_ini 1X: reserved
[15:14]	RW	vo_sd_clk_div	VO SD clock divider 00: divided by 2 01: divided by 4 1X: divided by 1
[13:12]	RW	vo_sd_clk_sel	VO SD clock select 00: clk_vo_sd_ini 01: clk_vo_hd0_ini 1X: reserved
[11:7]	RW	reserved	Reserved
[6]	RW	vdac_ch0_cken	VDAC channel 0 clock gating 0: disabled 1: enabled
[5]	RW	vo_hdate_cken	VO HDATE clock gating 0: disabled 1: enabled
[4]	RW	vo_hd_cken	VO HD clock gating 0: disabled 1: enabled
[3]	RW	vo_sdate_cken	VO SDATE clock gating 0: disabled 1: enabled
[2]	RW	vo_sd_cken	VO SD clock gating 0: disabled 1: enabled
[1]	RW	vo_cken	VO clock gating 0: disabled 1: enabled
[0]	RW	vo_bus_cken	VO bus clock gating 0: disabled 1: enabled

PERI_CRG60

PERI_CRG60 is a VPSS clock and soft reset control register.



Offset Address												Register Name												Total Reset Value											
0x00F0												PERI_CRG60												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												vpssclk_loaden	vpssclk_skipcfg				reserved	vpss_clk_sel	reserved		vpss_srst_req	reserved		vpss_cken										
Reset	0	0	0	0	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																															
[31:18]	RW	reserved		Reserved																															
[17]	RW	vpssclk_loaden		VPSS pulse filter/frequency scale enable 0: disabled 1: enabled																															
[16:12]	RW	vpssclk_skipcfg		VPSS pulse filter/frequency scale 0x0: No pulse is filtered out. 0x1: One pulse is filtered out. 0x2: Two pulses are filtered out. The rule applies to other values.																															
[11:10]	RW	reserved		Reserved																															
[9:8]	RW	vpss_clk_sel		VPSS clock select 00: 300 MHz 01: reserved 10: reserved 11: 400 MHz																															
[7:5]	RW	reserved		Reserved																															
[4]	RW	vpss_srst_req		VPSS soft reset 0: not reset 1: reset																															
[3:1]	RW	reserved		Reserved																															
[0]	RW	vpss_cken		VPSS clock gating 0: disabled 1: enabled																															



PERI_CRG63

PERI_CRG63 is a PVR-1 clock and soft reset control register.

	Offset Address																Register Name																Total Reset Value															
	0x00FC																PERI_CRG63																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																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	0	0	0	0	0	0	0	1	1	1	1	1	1			
Bits	Access	Name				Description																																										
[31:23]	RW	reserved				Reserved																																										
[22]	RW	pvr_srst_req				PVR soft reset 0: not reset 1: reset																																										
[21:17]	RW	reserved				Reserved																																										
[16]	RW	pvr_tsi2_pcrtl				PVR TSI2 phase select 0: normal phase 1: inverted																																										
[15]	RW	pvr_tsi1_pcrtl				PVR TSI1 phase select 0: normal phase 1: inverted																																										
[14:5]	RW	reserved				Reserved																																										
[4]	RW	pvr_tsi2_cken				PVR TSI2 clock gating 0: disabled 1: enabled																																										
[3]	RW	pvr_tsi1_cken				PVR TSI1 clock gating 0: disabled 1: enabled																																										
[2]	RW	pvr_27m_cken				PVR 27M clock gating 0: disabled 1: enabled																																										



[1]	RW	pvr_dmx_cken	PVR clock gating 0: disabled 1: enabled
[0]	RW	pvr_bus_cken	PVR bus clock gating 0: disabled 1: enabled

PERI_CRG64

PERI_CRG64 is a PVR-2 clock and soft reset control register.

	Offset Address 0x0100																Register Name PERI_CRG64								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																pvr_dmx_clkdiv_cfg		sw_dmxclk_loaden		sw_dmx_clk_div				pvr_dmx_clk_sel		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		
Bits	Access	Name	Description																													
[31:11]	RW	reserved	Reserved																													
[10]	RW	pvr_dmx_clkdiv_cfg	PVR DEMUX clock divider select 0: configured by software 1: configured by hardware																													
[9]	RW	sw_dmxclk_loaden	PVR pulse filter/frequency scale enable 0: disabled 1: enabled																													
[8:4]	RW	sw_dmx_clk_div	PVR pulse filter/frequency scale 0x0: No pulse is filtered out. 0x1: One pulse is filtered out. 0x2: Two pulses are filtered out. The rule applies to other values.																													



[3:2]	RW	pvr_dmx_clk_sel	PVR DEMUX clock select 00: 250 MHz 01: 288 MHz 1X: reserved
[1:0]	RW	reserved	Reserved

PERI_CRG67

PERI_CRG67 is a HDMITX_CTRL clock and soft reset control register.

	Offset Address 0x010C																Register Name PERI_CRG67								Total Reset Value 0x0000_003F							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																hdmitx_ctrl_asclk_sel	reserved	hdmitx_ctrl_cec_clk_sel	reserved	hdmitx_ctrl_srst_req	hdmitx_ctrl_bus_srst_req	reserved	reserved	hdmitx_ctrl_as_cken	hdmitx_ctrl_os_cken	hdmitx_ctrl_mhl_cken	hdmitx_ctrl_id_cken	hdmitx_ctrl_cec_cken	hdmitx_ctrl_bus_cken		
Reset	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																												
[31:15]	RW	reserved		Reserved																												
[14]	RW	hdmitx_ctrl_asclk_sel		HDMI TX ASCLK select 0: CRG (100 MHz) 1: HDMI TX PHY																												
[13]	RW	reserved		Reserved																												
[12]	RW	hdmitx_ctrl_cec_clk_sel		HDMI TX CEC clock select 0: 2 MHz 1: 2.02 MHz																												
[11:10]	RW	reserved		Reserved																												
[9]	RW	hdmitx_ctrl_srst_req		HDMI TX soft reset 0: not reset 1: reset																												
[8]	RW	hdmitx_ctrl_bus_srst_req		HDMI TX bus soft reset 0: not reset 1: reset																												
[7:6]	RW	reserved		Reserved																												



[5]	RW	hdmitx_ctrl_as_cke_n	HDMI TX AS clock gating 0: disabled 1: enabled
[4]	RW	hdmitx_ctrl_os_cke_n	HDMI TX OS clock gating 0: disabled 1: enabled
[3]	RW	hdmitx_ctrl_mhl_cken	HDMI TX clock gating 0: disabled 1: enabled
[2]	RW	hdmitx_ctrl_id_cke_n	HDMI TX ID clock gating 0: disabled 1: enabled
[1]	RW	hdmitx_ctrl_cec_cken	HDMI TX CEC clock gating 0: disabled 1: enabled
[0]	RW	hdmitx_ctrl_bus_cken	HDMI TX bus clock gating 0: disabled 1: enabled

PERI_CRG68

PERI_CRG68 is a HDMITX_PHY clock and soft reset control register.

	Offset Address																Register Name								Total Reset Value								
	0x0110																PERI_CRG68								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																										hdmitx_phy_srst_req	reserved	hdmitx_phy_bus_cken				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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]	RW	reserved				Reserved																											



[4]	RW	hdmitx_phy_srst_req	HDMI TX PHY soft reset 0: not reset 1: reset
[3:1]	RW	reserved	Reserved
[0]	RW	hdmitx_phy_bus_cken	HDMI TX PHY bus clock gating 0: disabled 1: enabled

PERI_CRG69

PERI_CRG69 is an ADAC clock and soft reset control register.

	Offset Address 0x0114																Register Name PERI_CRG69								Total Reset Value 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																									adac_srst_req	reserved	adac_cken				
Reset	0	0	0	0	0	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]	RW	reserved		Reserved																												
[4]	RW	adac_srst_req		ADAC soft reset 0: not reset 1: reset																												
[3:1]	RW	reserved		Reserved																												
[0]	RW	adac_cken		ADAC bus clock gating 0: disabled 1: enabled																												

PERI_CRG70

PERI_CRG70 is an AIAO clock and soft reset control register.



Offset Address												Register Name												Total Reset Value											
0x0118												PERI_CRG70												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												aiao_mclk_sel	reserved	aiaoclk_loaden	aiaoclk_skipcfg	reserved				aiao_srst_req	reserved				aiao_cken									
Reset	0	0	0	0	0	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																														
[31:22]	RW	reserved			Reserved																														
[21:20]	RW	aiao_mclk_sel			AIAO MCLK source select 00: 1000 MHz 01: 1200 MHz 1X: 1500 MHz																														
[19:18]	RW	reserved			Reserved																														
[17]	RW	aiaoclk_loaden			AIAO pulse filter/frequency scale enable 0: disabled 1: enabled																														
[16:12]	RW	aiaoclk_skipcfg			AIAO pulse filter/frequency scale 0x0: No pulse is filtered out. 0x1: One pulse is filtered out. 0x2: Two pulses are filtered out. The rule applies to other values.																														
[11:5]	RW	reserved			Reserved																														
[4]	RW	aiao_srst_req			AIAO soft reset 0: not reset 1: reset																														
[3:1]	RW	reserved			Reserved																														
[0]	RW	aiao_cken			AIAO clock gating 0: disabled 1: enabled																														

PERI_CRG71

PERI_CRG71 is a VDAC clock and soft reset control register.



	Offset Address 0x011C																Register Name PERI_CRG71																Total Reset Value 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																vdac_c_clk_pctrl	reserved																vdac_chop_cken		
Reset	0	0	0	0	0	0	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]	RW	reserved				Reserved																														
[16]	RW	vdac_c_clk_pctrl				VDAC C phase select 0: normal phase 1: inverted																														
[15:1]	RW	reserved				Reserved																														
[0]	RW	vdac_chop_cken				VDAC chop clock gating 0: disabled 1: enabled																														

PERI_CRG72

PERI_CRG72 is an FE PHY clock and soft reset control register.

	Offset Address 0x0120																Register Name PERI_CRG72																Total Reset Value 0x0000_0011			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved																fephy_clk_sel	reserved				fephy_srst_req	reserved				fephy_cken									
Reset	0	0	0	0	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	1			
Bits	Access	Name				Description																														
[31:9]	RW	reserved				Reserved																														



[8]	RW	fephy_clk_sel	FE PHY clock select 0: 25 MHz 1: 40 MHz
[7:5]	RW	reserved	Reserved
[4]	RW	fephy_srst_req	FE PHY soft reset 0: not reset 1: reset
[3:1]	RW	reserved	Reserved
[0]	RW	fephy_cken	FE PHY clock gating 0: disabled 1: enabled

PERI_CRG73

PERI_CRG73 is a GPU_LP clock and soft reset control register.

	Offset Address 0x0124																Register Name PERI_CRG73																Total Reset Value 0x0000_0206															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name	reserved																													gpu_freq_sel_cfg_crg																		
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	1	1	0																
Bits	Access	Name			Description																																											
[31:11]	RW	reserved			Reserved																																											
[10]	RW	gpu_sw_begin_cfg			GPU switch start 0: not started 1: started																																											
[9]	RW	gpu_begin_cfg_bypass			GPU switch start bypass signal 0: signal output by the state machine 1: signal configured by registers																																											
[8:3]	RW	reserved			Reserved																																											



[2:0]	RW	gpu_freq_sel_cfg_crg	GPU clock select 000: 432 MHz 001: 400 MHz 010: 375 MHz 011: 345.6 MHz 100: 300 MHz 101: 250 MHz 110: 200 MHz 111: 500 MHz
-------	----	----------------------	--

PERI_CRG74

PERI_CRG74 is a DDR_LP clock and soft reset control register.

	Offset Address 0x0128																Register Name PERI_CRG74								Total Reset Value 0x0000_0204							
Bit	31	30	29	28	27	26	25	24	23	22	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_sw_begin_cfg		ddr_begin_cfg_bypass		reserved				ddr_freq_sel_cfg_crg							
Reset	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	1	0	0		
Bits	Access	Name		Description																												
[31:11]	RW	reserved		Reserved																												
[10]	RW	ddr_sw_begin_cfg		DDR switch start 0: not started 1: started																												
[9]	RW	ddr_begin_cfg_bypass		DDR switch start bypass signal 0: signal output by the state machine 1: signal configured by registers																												
[8:3]	RW	reserved		Reserved																												



[2:0]	RW	ddr_freq_sel_cfg_crg	DDR clock select 000: DPLL output 001: DPLL output 010: DPLL output 011: DPLL output 100: 24 MHz 101: 400 MHz 110: 300 MHz 111: 200 MHz
-------	----	----------------------	---

PERI_CRG85

PERI_CRG85 is an SW_READBACK register.

Offset Address				Register Name																Total Reset Value												
Bit	0x0154			PERI_CRG85																0x0000_0000												
Name	31	30	29	28	27	26	25	24	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	
Bits				Access				Name				Description																				
[31]	RO			reserved				Reserved																								
[30:29]	RO			vdh_clk_seled				VDH clock switch completion indicator																								
[28]	RO			reserved				Reserved																								
[27:26]	RO			mde3_clk_seled				MDE3 clock switch completion indicator																								
[25:24]	RO			mde2_clk_seled				MDE2 clock switch completion indicator																								
[23:22]	RO			mde1_clk_seled				MDE1 clock switch completion indicator																								
[21:20]	RO			mde0_clk_seled				MDE0 clock switch completion indicator																								
[19:18]	RO			sdio1_clk_seled				SDIO1 clock switch completion indicator																								
[17:16]	RO			sdio0_clk_seled				SDIO0 clock switch completion indicator																								
[15:14]	RO			reserved				Reserved																								
[13:12]	RO			core_bus_clk_seled				Bus clock switch completion indicator																								



[11:9]	RO	ddr_clk_mux	DDR clock switch completion indicator
[8]	RO	ddr_clk_sw_ok_crg	DDR PLL switch completion indicator
[7:5]	RO	gpu_clk_mux	GPU clock switch completion indicator
[4]	RO	gpu_clk_sw_ok_cr_g	GPU PLL switch completion indicator
[3:1]	RO	cpu_clk_mux	CPU clock switch completion indicator
[0]	RO	cpu_clk_sw_ok_cr_g	CPU PLL switch completion indicator

PERI_CRG90

PERI_CRG90 is an output reset control register.

	Offset Address 0x0168																Register Name PERI_CRG90								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									mute_ctrl_out	dem_RST0_out	reserved	slic_RST_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		
Bits	Access	Name		Description																												
[31:4]	RW	reserved		Reserved																												
[3]	RW	mute_ctrl_out		Soft reset request of the off-chip MUTE pin 0: muted 1: not muted																												
[2]	RW	dem_RST0_out		Soft reset request for off-chip demo 0 0: reset deasserted 1: reset																												
[1]	RW	reserved		Reserved																												
[0]	RW	slic_RST_out		Soft reset request for off-chip SLIC 0: reset deasserted 1: reset																												



PERI_CRG91

PERI_CRG91 is a USB_FREECLK_DEC control register.

	Offset Address	Register Name	Total Reset Value
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	PERI_CRG91	0x0000_0000
Name		reserved	usb3_phy_cnt_out usb2_phy2_cnt_out usb2_phy_cnt_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
[31:3]	RO	reserved	Reserved
[2]	RO	usb3_phy_cnt_out	Readback of the USB3 PHY FREECLK count 0: FREECLK is not output. 1: FREECLK is output.
[1]	RO	usb2_phy2_cnt_out	Readback of the USB2 PHY2 FREECLK count 0: FREECLK is not output. 1: FREECLK is output.
[0]	RO	usb2_phy_cnt_out	Readback of the USB2 PHY FREECLK count 0: FREECLK is not output. 1: FREECLK is output.

PERI_CRG93

PERI_CRG93 is a VDH_RST_READBACK register.



	Offset Address																Register Name								Total Reset Value							
	0x0174																PERI_CRG93								0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									vdh_mfd_RST_OK	vdh_scd_RST_OK	vdh_all_RST_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		
Bits	Access	Name		Description																												
[31:3]	RO	reserved		Reserved																												
[2]	RO	vdh_mfd_RST_OK		MFD reset status 0: reset deasserted 1: reset																												
[1]	RO	vdh_scd_RST_OK		SCD reset status 0: reset deasserted 1: reset																												
[0]	RO	vdh_all_RST_OK		VDH global reset status 0: reset deasserted 1: reset																												

PERI_CRG94

PERI_CRG94 is a WDG clock and soft reset control register.

	Offset Address																Register Name								Total Reset Value							
	0x0178																PERI_CRG94								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																										wdg0_SRST_REQ	reserved	wdg0_CKEN			
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RW	reserved	Reserved
[4]	RW	wdg0_srst_req	WDG0 soft reset 0: not reset 1: reset
[3:1]	RW	reserved	Reserved
[0]	RW	wdg0_cken	WDG0 clock gating 0: disabled 1: enabled

PERI_CRG95

PERI_CRG95 is a PLL_TEST clock and soft reset control register.

	Offset Address 0x017C																Register Name PERI_CRG95								Total Reset Value 0x0000_0005							
Bit	31	30	29	28	27	26	25	24	23	22	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_clk_en	test_clk_sel	pll_test_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	1	0	1	
Bits	Access	Name		Description																												
[31:3]	RW	reserved		Reserved																												
[2]	RW	test_clk_en		clk_test_out output enable 0: disabled 1: enabled																												
[1]	RW	test_clk_sel		clk_test_out source select 0: CPU clock divided by 32 1: DDR clock divided by 16																												
[0]	RW	pll_test_en		PLL test enable 0: disabled 1: enabled																												

PERI_CRG96

PERI_CRG96 is a CPU RAM speed control register.



Offset Address								Register Name								Total Reset Value																
0x0180								PERI_CRG96								0x0000_A501																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved												mem_adjust_cpu																			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	1	
Bits	Access	Name			Description																											
[31:19]	RW	reserved			Reserved																											
[18:0]	RW	mem_adjust_cpu			l2_dataram_delay[18:14] speed adjustment parameter a7_ram_delay[13:0] speed adjustment parameter																											

PERI_CRG97

PERI_CRG97 is a CPU RAM speed control register.

Offset Address								Register Name								Total Reset Value																
0x0184								PERI_CRG97								0x0000_4212																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																mem_adjust_gpu															
Reset	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	0	1	0	0	1	0	
Bits	Access	Name			Description																											
[31:16]	RW	reserved			Reserved																											
[15:0]	RW	mem_adjust_gpu			ema_rfs[15:13] speed adjustment parameter emaw_rfs[12:11] speed adjustment parameter ema_ras[10:8] speed adjustment parameter emaw_ras[7:6] speed adjustment parameter emaa_rft[5:3] speed adjustment parameter emab_rft[2:0] speed adjustment parameter																											

PERI_CRG100

PERI_CRG100 is a USB2 PHY2 clock and soft reset control register.



Offset Address								Register Name								Total Reset Value																
0x0190								PERI_CRG100								0x0000_0301																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								usb2_phy2_refclk_sel								reserved								usb2_phy2_ref_cken							
Reset	0	0	0	0	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	0	
Bits	Access	Name		Description																												
[31:17]	RW	reserved		Reserved																												
[16]	RW	usb2_phy2_refclk_sel		USB2 PHY reference clock select 0: crystal oscillator clock 1: 24 MHz internal clock																												
[15:10]	RW	reserved		Reserved																												
[9]	RW	usb2_phy2_srst_treq		USB2 PHY TPOR soft reset 0: not reset 1: reset																												
[8]	RW	usb2_phy2_srst_req		USB2 PHY POR soft reset 0: not reset 1: reset																												
[7:1]	RW	reserved		Reserved																												
[0]	RW	usb2_phy2_ref_cken		USB2 PHY reference clock gating 0: disabled 1: enabled																												

PERI_CRG101

PERI_CRG101 is a USB3 PHY clock and soft reset control register.



Offset Address																Register Name								Total Reset Value								
0x0194																PERI_CRG101								0x0000_0431								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																usb3_ref_ssp_en	usb3_ref_use_pad	usb3_phy_refclk_sel	reserved	usb3_phy_srst_treq	usb3_phy_srst_req	reserved	usb3_phy_ref_cken								
Reset	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	1	1	0	0	0	1	
Bits	Access	Name		Description																												
[31:11]	RW	reserved		Reserved																												
[10]	RW	usb3_ref_ssp_en		USB3 PHY super-speed domain clock enable 0: disabled 1: enabled																												
[9]	RW	usb3_ref_use_pad		USB3 PHY reference clock select 0: internal CRG 1: pad																												
[8]	RW	usb3_phy_refclk_sel		USB3 PHY reference clock select 0: crystal oscillator clock 1: 24 MHz internal clock																												
[7:6]	RW	reserved		Reserved																												
[5]	RW	usb3_phy_srst_treq		USB3 PHY TPOR soft reset 0: not reset 1: reset																												
[4]	RW	usb3_phy_srst_req		USB3 PHY POR soft reset 0: not reset 1: reset																												
[3:1]	RW	reserved		Reserved																												
[0]	RW	usb3_phy_ref_cken		USB3 PHY reference clock gating 0: disabled 1: enabled																												

PERI_CRG102

PERI_CRG102 is a USB2 CTRL2 clock and soft reset control register.



Offset Address								Register Name								Total Reset Value																
0x0198								PERI_CRG102								0x0011_5037																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								usb2_clk48_sel1								reserved								usb2_utmi0_cken1							
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	1	1	0	1	1	1	
Bits	Access	Name		Description																												
[31:21]	RW	reserved		Reserved																												
[20]	RW	usb2_clk48_sel1		USB2 CLK48 clock source 0: provided by the CRG 1: PHY output																												
[19:17]	RW	reserved		Reserved																												
[16]	RW	usb2_hst_phy_srst_req1		USB2 CTRL hst_phy soft reset 0: not reset 1: reset																												
[15]	RW	reserved		Reserved																												
[14]	RW	usb2_utmi0_srst_req1		USB2 CTRL UTMI0 soft reset 0: not reset 1: reset																												
[13]	RW	reserved		Reserved																												
[12]	RW	usb2_bus_srst_req1		USB2 CTRL bus soft reset 0: not reset 1: reset																												
[11:6]	RW	reserved		Reserved																												
[5]	RW	usb2_utmi0_cken1		USB2 CTRL UTMI0 clock gating 0: disabled 1: enabled																												
[4]	RW	usb2_hst_phy_cken1		USB2 CTRL hst_phy clock gating 0: disabled 1: enabled																												
[3]	RW	reserved		Reserved																												



[2]	RW	usb2_ohci12m_cken1	USB2 CTRL OHCI12M clock gating 0: disabled 1: enabled
[1]	RW	usb2_ohci48m_cken1	USB2 CTRL OHCI48M clock gating 0: disabled 1: enabled
[0]	RW	usb2_bus_cken1	USB2 CTRL bus clock gating 0: disabled 1: enabled

3.4 System Controller

3.4.1 Overview

The system controller controls the operating mode of the system, monitors the system status, manages the key functions of the system (such as the clock, reset, and pin multiplexing), and configures some functions of peripherals.

3.4.2 Features

The system controller has the following features:

- Controls and monitors the operating mode of the system.
- Controls the system clock and queries its status.
- Controls system address remapping and monitors the remapping status.
- Provides common peripheral registers.
- Controls pin multiplexing of the always-on area.
- Provides write protection for key registers.
- Provides chip ID registers.

3.4.3 Function Description

Operating Modes

The system supports three operating modes:

- Normal mode

Typically, the system works in normal mode. In this mode, the system is driven by the output clock of the on-chip PLL. All modules can work properly under this clock source.

- Slow mode

The slow mode is a low-speed mode. In this mode, the system is driven by the external crystal oscillator clock, and only some on-chip peripherals (such as the system controller, timer, and NANDC) can work. Modules (such as the DDRC) that require high-speed clocks cannot work.



- Doze mode

The doze mode is also a low-speed mode, but its speed is lower than that of the slow mode. Only a few on-chip peripherals can work in doze mode. In this mode, the system is driven by a 100 kHz low-frequency clock generated by dividing the frequency of the external crystal oscillator clock. Except the CPU and a few modules (such as the system controller, timer, IR, and UART1), most on-chip peripherals and the memory interfaces cannot work in this mode.

The always-on area can work in any of the preceding modes, whereas the power-off area can work only in normal or slow mode.

When the system operating mode is switched [PERI_CRG22](#), [PERI_CRG73](#), [PERI_CRG74](#), and [SC_CTRL](#) need to be configured to select working clocks.

The operating mode of the always-on area is switched by configuring [SC_CTRL\[mcu_bus_clk_sel\]](#) for switching the system clock source. These two bits define the target operating mode to be entered:

- 00: slow mode
- 01: normal mode
- 10: doze mode
- 11: reserved

The operating mode of the power-off area is switched by configuring [PERI_CRG22](#), [PERI_CRG73](#), and [PERI_CRG74](#) for selecting clock frequencies.

Soft Reset

- The system controller can soft-reset the entire chip or some modules.
- After the global soft reset register [SC_SYSRES](#) is configured, the system controller sends a reset request to the on-chip reset module. Then the chip is reset.
- After the soft reset bit of a module in the system controller is configured, the system controller instructs the on-chip reset module to reset this module.
- After a request for resetting the power-off area is configured in the system controller, the entire power-off area is reset.

System Address Remapping

The system controller supports address remapping and allows the address decoding unit to remap and reallocate the system storage address space. This section uses the mode of booting the system from the on-chip ROM as an example.

After POR, the system addresses are remapped. The addresses ranging from 0xFFFF_0000 to 0xFFFF_FFFF are mapped to the on-chip ROM. After the address remap is cleared by configuring [SC_CTRL\[remapclear\]](#), the addresses are mapped to the on-chip SRAM.

Pin Multiplexing Control

The system controller supports pin multiplexing control for the always-on area. For details, see section [3.4.5 "Register Description."](#)



Write Protection for Key Registers

To prevent the entire system from being affected by misoperations on the system controller, the system controller provides write protection for the following key configuration registers:

- Mode switching control register [SC_CTRL](#)
- System soft reset control register [SC_SYSRES](#)

Before writing to these key registers, you must disable the write protection function by configuring [SC_LOCKEN](#). After writes, you need to enable the write protection function by configuring [SC_LOCKEN](#) again so that these key registers are not overwritten by the software arbitrarily.



NOTE
Write protection is not enabled for key registers after reset by default. You are advised to enable write protection by configuring SC_LOCKEN when the system starts.

Chip Flag Register

The system controller provides a chip ID register [SC_SYSID](#). It is a 32-bit read-only identifier register.

3.4.4 Register Summary

[Table 3-44](#) describes SYS_CTRL registers.

Table 3-44 Summary of SYS_CTRL registers (base address: 0xF800_0000)

Offset Address	Register	Description	Page
0x0000	SC_CTRL	System control register	3-75
0x0004	SC_SYSRES	System status register	3-76
0x0040	SC_LOW_POWER_CTRL	Low-power control register	3-76
0x0044	SC_IO_REUSE_SEL	MCU pin multiplexing control register	3-78
0x0048	SC_CLKGATE_SRST_CTRL	MCU internal module clock gating and soft reset control register	3-80
0x0050	SC_WDG_RST_CTRL	Software-controlled WDG reset register	3-81
0x0058	SC_DDRPHY_LP_EN	DDR PHY low-power control register	3-82
0x0060	SC MCU_HPM_CTRL	HPM control register	3-82
0x0064	SC MCU_HPM_STAT	HPM status register	3-83
0x0068	SC MCU_LDO_CTRL	Low-dropout (LDO) control register	3-83
0x0080	SC_GEN0	System common register 0	3-84
0x0084	SC_GEN1	System common register 1	3-84
0x0088	SC_GEN2	System common register 2	3-84



Offset Address	Register	Description	Page
0x008C	SC_GEN3	System common register 3	3-85
0x0090	SC_GEN4	System common register 4	3-85
0x0094	SC_GEN5	System common register 5	3-85
0x0098	SC_GEN6	System common register 6	3-86
0x009C	SC_GEN7	System common register 7	3-86
0x00A0	SC_GEN8	System common register 8	3-86
0x00A4	SC_GEN9	System common register 9	3-87
0x00A8	SC_GEN10	System common register 10	3-87
0x00AC	SC_GEN11	System common register 11	3-87
0x00B0	SC_GEN12	System common register 12	3-88
0x00B4	SC_GEN13	System common register 13	3-88
0x00B8	SC_GEN14	System common register 14	3-88
0x00BC	SC_GEN15	System common register 15	3-89
0x00C0	SC_GEN16	System common register 16	3-89
0x00C4	SC_GEN17	System common register 17	3-89
0x00C8	SC_GEN18	System common register 18	3-90
0x00CC	SC_GEN19	System common register 19	3-90
0x00D0	SC_GEN20	System common register 20	3-90
0x00D4	SC_GEN21	System common register 21	3-91
0x00D8	SC_GEN22	System common register 22	3-91
0x00DC	SC_GEN23	System common register 23	3-91
0x00E0	SC_GEN24	System common register 24	3-92
0x00E4	SC_GEN25	System common register 25	3-92
0x00E8	SC_GEN26	System common register 26	3-93
0x00EC	SC_GEN27	System common register 27	3-93
0x00F0	SC_GEN28	System common register 28	3-93
0x00F4	SC_GEN29	System common register 29	3-94
0x00F8	SC_GEN30	System common register 30	3-94
0x00FC	SC_GEN31	System common register 31	3-94
0x0100	SC_GPIO_OD_CTRL	GPIO open drain (OD) control register	3-95



Offset Address	Register	Description	Page
0x020C	SC_LOCKEN	Write protection control register	3-95
0x0EE0	SC_SYSID	Chip ID register	3-96
0x0F00	SC_GEN32	System common register 32	3-96
0x0F04	SC_GEN33	System common register 33	3-97
0x0F08	SC_GEN34	System common register 34	3-97
0x0F0C	SC_GEN35	System common register 35	3-98
0x0F10	SC_GEN36	System common register 36	3-98
0x0F14	SC_GEN37	System common register 37	3-98
0x0F18	SC_GEN38	System common register 38	3-99
0x0F1C	SC_GEN39	System common register 39	3-99
0x0F20	SC_GEN40	System common register 40	3-99
0x0F24	SC_GEN41	System common register 41	3-100
0x0F28	SC_GEN42	System common register 42	3-100
0x0F2C	SC_GEN43	System common register 43	3-100
0x0F30	SC_GEN44	System common register 44	3-101
0x0F34	SC_GEN45	System common register 45	3-101
0x0F38	SC_GEN46	System common register 46	3-101
0x0F3C	SC_GEN47	System common register 47	3-102
0x0F40	SC_GEN48	System common register 48	3-102
0x0F44	SC_GEN49	System common register 49	3-102
0x0F48	SC_GEN50	System common register 50	3-103
0x0F4C	SC_GEN51	System common register 51	3-103
0x0F50	SC_GEN52	System common register 52	3-103
0x0F54	SC_GEN53	System common register 53	3-104
0x0F58	SC_GEN54	System common register 54	3-104
0x0F5C	SC_GEN55	System common register 55	3-104
0x0F60	SC_GEN56	System common register 56	3-105
0x0F64	SC_GEN57	System common register 57	3-105
0x0F68	SC_GEN58	System common register 58	3-105
0x0F6C	SC_GEN59	System common register 59	3-106



Offset Address	Register	Description	Page
0x0F70	SC_GEN60	System common register 60	3-106
0x0F74	SC_GEN61	System common register 61	3-106
0x0F78	SC_GEN62	System common register 62	3-107
0x0F7C	SC_GEN63	System common register 63	3-107

3.4.5 Register Description

SC_CTRL

SC_CTRL is a system control register. It is used to specify the operations to be performed by the system.

	Offset Address	Register Name	Total Reset Value
	0x0000	SC_CTRL	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	remapstat remapclear reserved mcu_bus_clk_div mcu_bus_clk_sele_stat mcu_bus_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 1 0 0 0 0 0 0 0 0 0		
Bits	Access	Name	Description
[31:10]	RW	reserved	Reserved Reading this field returns 0 and writing to this field has no effect.
[9]	RO	remapstat	Address remap status 0: remap clear 1: remap
[8]	RW	remapclear	Address remap clear 0: not cleared 1: cleared
[7:6]	RW	reserved	Reserved Reading this field returns 0 and writing this field has no effect.



[5:4]	RW	mcu_bus_clk_div	MCU bus clock mcu_bus_clk frequency divider 00: divided by 1 01: divided by 2 10: divided by 3 11: divided by 4
[3:2]	RO	mcu_bus_clk_sele_stat	MCU bus clock status 00: 24 MHz 01: 200 MHz 10: 100 kHz 11: reserved
[1:0]	RW	mcu_bus_clk_sel	MCU bus clock select 00: 24 MHz 01: 200 MHz 10: 100 kHz 11: reserved

SC SYSRES

SC_SYSRES is a system status register. When a 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.

SC_LOW_POWER_CTRL

SC LOW POWER CTRL is a low-power control register.



	Offset Address	Register Name	Total Reset Value
Bit	0x0040	SC_LOW_POWER_CTRL	0x0000_0402
Name	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	reserved	rng_mcu_flag core_pwr_active bus_core_pd_idleack bus_core_pd_idle bus_core_pd_idlereq reserved mcu_lp_subsys_iso reserved stb_poweroff reserved
Reset	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 1 0		
Bits	Access	Name	Description
[31:11]	RO	reserved	Reserved
[10]	RO	rng_mcu_flag	Whether the random numbers of the random number generator (RNG) in the always-on area are latched 0x0: no 0x1: yes
[9]	RO	core_pwr_active	Whether the MCU has accessed the powered-off core area 0: no 1: yes
[8]	RO	bus_core_pd_idleack	Power-down acknowledgement
[7]	RO	bus_core_pd_idle	Power-down completion
[6]	RW	bus_core_pd_idlereq	Power-down request
[5:4]	RW	reserved	Reserved
[3]	RW	mcu_lp_subsys_iso	Isolation area control 0: disabled. That is, signals are not isolated. 1: enabled. That is, signals are isolated.
[2]	RW	reserved	Reserved
[1]	RW	stb_poweroff	Polarity of the STANDBY_PWROFF pin 0: low-level output 1: high-level output
[0]	RW	reserved	Reserved



SC_IO_REUSE_SEL

SC_IO_REUSE_SEL is an MCU subsystem pin multiplexing control register.

Offset Address												Register Name												Total Reset Value											
0x0044												SC_IO_REUSE_SEL												0x0038_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	reserved												sim0_det_padctrl_PD	sim0_data_padctrl_PD	sim0_clk_padctrl_PD	sim0_clk_padctrl_DS	sim0_clk_padctrl_SR	reserved	led_data_padctrl_PD	led_clk_padctrl_PD	reserved	stb_gpio_sel	reserved	ir_gpio_sel	csn0_gpio_sel	reserved	data_gpio_sel	clk_gpio_sel	reserved	uart_txd_gpio_sel	uart_rxd_gpio_sel				
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
Bits	Access	Name			Description																														
[31:25]	RW	reserved			Reserved																														
[24]	RW	sim0_det_padctrl_PD			SIM0_DET pull-down control enable 0: disabled 1: enabled																														
[23]	RW	sim0_data_padctrl_PD			SIM0_DATA pull-down control enable 0: disabled 1: enabled																														
[22]	RW	sim0_clk_padctrl_PD			SIM0_CLK pull-down control enable 0: disabled 1: enabled																														
[21:20]	RW	sim0_clk_padctrl_DS			SIM0_CLK drive current 00: 4 mA 01: 3 mA 10: 2 mA 11: 1 mA When this pin is used as the SIM0_CLK function, the value 01 is recommended in OD output mode, and the value 10 is recommended in CMOS output mode. When this pin is used as other functions, the value 11 is recommended.																														
[19]	RW	sim0_clk_padctrl_SR			SIM0_CLK SR control enable 0: enabled 1: disabled (recommended)																														



[18]	RW	reserved	Reserved
[17]	RW	led_data_padctrl_PD	LED_DATA pull-down control enable 0: disabled 1: enabled This bit must be set to 1 for the QFP216 32-bit DDR.
[16]	RW	led_clk_padctrl_PD	LED_CLK pull-down control enable 0: disabled 1: enabled This bit must be set to 1 for the QFP216 32-bit DDR.
[15:14]	RW	reserved	Reserved
[13]	RW	stb_gpio_sel	Multiplexing control for the STANDBY_PWROFF pin 0: STANDBY_PWROFF 1: GPIO5_0
[12:11]	RW	reserved	Reserved
[10]	RW	ir_gpio_sel	Multiplexing control for the IR_IN pin 0: IR_IN 1: GPIO5_1
[9:8]	RW	csn0_gpio_sel	Multiplexing control for the LED_CSN0 pin 00: LED_KEY0 01: GPIO5_2 10: RSTN_IN
[7:6]	RW	reserved	Reserved
[5]	RW	data_gpio_sel	Multiplexing control for the LED_DATA pin 0: GPIO5_5 1: LED DATA
[4]	RW	clk_gpio_sel	Multiplexing control for the LED_CLK pin 0: GPIO5_6 1: LED CLK
[3:2]	RW	reserved	Reserved
[1]	RW	uart_txd_gpio_sel	Multiplexing control for the UART0_TXD pin 0: UART0_TXD 1: UART1_TXD
[0]	RW	uart_rxd_gpio_sel	Multiplexing control for the UART0_RXD pin 0: UART0_RXD 1: UART1_RXD



SC_CLKGATE_SRST_CTRL

SC_CLKGATE_SRST_CTRL is an MCU internal module clock gating and soft reset control register.

		Offset Address 0x0048																Register Name SC_CLKGATE_SRST_CTRL																Total Reset Value 0x0001_1151															
Bit	Name	31	30	29	28	27	26	25	24	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	1	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	0	1																
		reserved																uart_srst_req																reserved															
Bits		Access		Name				Description																																									
[31:29]		RW		reserved				Reserved																																									
[28]		RW		pd_rst_req				Power-down reset request 0: not reset 1: reset																																									
[27:14]		RW		reserved				Reserved																																									
[13]		RW		uart_srst_req				UART soft reset 0: not reset 1: reset																																									
[12]		RW		uart_cken				UART clock gating 0: disabled 1: enabled																																									
[11:10]		RW		reserved				Reserved																																									
[9]		RW		led_srst_req				LED module soft reset 0: not reset 1: reset																																									
[8]		RW		led_cken				LEDC clock gating 0: disabled 1: enabled																																									
[7]		RW		timer01_srst_req				Timer 0 and timer 1 soft reset 0: not reset 1: reset																																									



[6]	RW	timer01_cken	Timer 0 and timer 1 clock gating 0: disabled 1: enabled
[5]	RW	ir_srst_req	IR module soft reset 0: not reset 1: reset
[4]	RW	ir_cken	IR clock gating 0: disabled 1: enabled
[3:2]	RW	reserved	Reserved
[1]	RW	mce_srst_req	MCU soft reset 0: not reset 1: reset
[0]	RW	mce_cken	MCU clock gating 0: disabled 1: enabled

SC_WDG_RST_CTRL

SC_WDG_RST_CTRL is a software-controlled WDG reset register.

	Offset Address 0x0050																Register Name SC_WDG_RST_CTRL								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																												wdg_rst_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		
Bits	Access	Name			Description																											
[31:1]	RW	reserved			Reserved																											
[0]	RW	wdg_rst_ctrl			WDG_RST reset signal (controlled by software). 0: not reset 1: enable WDG_RST software control. When timer 0 or timer 1 generates an interrupt, a reset signal is sent through the WDG_RST pin.																											



SC_DDRPHY_LP_EN

SC_DDRPHY_LP_EN is a DDR PHY low-power control register.

	Offset Address																Register Name								Total Reset Value									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	reserved																																	ddrphy_lp_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				
Bits	Access	Name		Description																														
[31:2]	RW	reserved		Reserved																														
[1:0]	RW	ddrphy_lp_en		DDR PHY mode 01: The DDR PHY is in normal mode. Other values: The DDR PHY is in retention mode.																														

SC MCU_HPM_CTRL

SC_MCU_HPM_CTRL is an HPM control register.

	Offset Address																Register Name								Total Reset Value							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																												mcu_hpm_en	mcu_hpm_div		
Reset	0	0	0	0	0	0	0	0	0	0	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	reserved		Reserved																												
[7]	RW	mcu_hpm_rst_req		HPM reset request control 0: no reset request 1: valid reset request																												



[6]	RW	mcu_hpm_en	HPM enable 0: disabled 1: enabled
[5:0]	RW	mcu_hpm_div	Frequency divider of the HPM working clock

SC MCU HPM STAT

SC MCU HPM STAT is an HPM status register.

SC MCU LDO CTRL

SC MCU LDO CTRL is an LDO control register.

Offset Address								Register Name								Total Reset Value																
0x0068								SC MCU LDO_CTRL								0x0000_0008																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									mcu_ldo_vset						
Reset	0	0	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		
Bits	Access	Name			Description																											
[31:4]	RW	reserved			Reserved																											



[3:0] RW mcu_ldo_vset mcu_ldo_vset value

SC_GEN0

SC_GEN0 is system common register 0.

SC GEN1

SC_GEN1 is system common register 1.

Offset Address								Register Name								Total Reset Value																
0x0084								SC_GEN1								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen1		This register is used to save the system status as required.																												

SC GEN2

SC_GEN2 is system common register 2.



Offset Address																Register Name								Total Reset Value								
0x0088																SC_GEN2								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																	sc_gen2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen2				This register is used to save the system status as required.																										

SC_GEN3

SC_GEN3 is system common register 3.

Offset Address																Register Name								Total Reset Value								
0x008C																SC_GEN3								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																	sc_gen3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen3				This register is used to save the system status as required.																										

SC_GEN4

SC_GEN4 is system common register 4.

Offset Address																Register Name								Total Reset Value								
0x0090																SC_GEN4								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																	sc_gen4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen4				This register is used to save the system status as required.																										

SC_GEN5

SC_GEN5 is system common register 5.



Offset Address			Register Name																Total Reset Value														
Bit	0x0094			SC_GEN5																0x0000_0000													
Name	sc_gen5																																
Reset	0 0 0 0 0 0 0 0																																
Bits	Access	Name																Description															
[31:0]	RW	sc_gen5																This register is used to save the system status as required.															

SC_GEN6

SC_GEN6 is system common register 6.

Offset Address			Register Name																Total Reset Value														
Bit	0x0098			SC_GEN6																0x0000_0000													
Name	sc_gen6																																
Reset	0 0 0 0 0 0 0 0																																
Bits	Access	Name																Description															
[31:0]	RW	sc_gen6																This register is used to save the system status as required.															

SC_GEN7

SC_GEN7 is system common register 7.

Offset Address			Register Name																Total Reset Value														
Bit	0x009C			SC_GEN7																0x0000_0000													
Name	sc_gen7																																
Reset	0 0 0 0 0 0 0 0																																
Bits	Access	Name																Description															
[31:0]	RW	sc_gen7																This register is used to save the system status as required.															

SC_GEN8

SC_GEN8 is system common register 8.



Offset Address																Register Name								Total Reset Value								
0x00A0																SC_GEN8								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen8																sc_gen8															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen8				This register is used to save the system status as required.																										

SC_GEN9

SC_GEN9 is system common register 9.

Offset Address																Register Name								Total Reset Value								
0x00A4																SC_GEN9								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen9																sc_gen9															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen9				This register is used to save the system status as required.																										

SC_GEN10

SC_GEN10 is system common register 10.

Offset Address																Register Name								Total Reset Value								
0x00A8																SC_GEN10								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen10																sc_gen10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen10				This register is used to save the system status as required.																										

SC_GEN11

SC_GEN11 is system common register 11.



Offset Address			Register Name																Total Reset Value													
0x00AC			SC_GEN11																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen11																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen11		This register is used to save the system status as required.																												

SC_GEN12

SC_GEN12 is system common register 12.

Offset Address			Register Name																Total Reset Value													
0x00B0			SC_GEN12																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen12																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen12		This register is used to save the system status as required.																												

SC_GEN13

SC_GEN13 is system common register 13.

Offset Address			Register Name																Total Reset Value													
0x00B4			SC_GEN13																													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen13																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen13		This register is used to save the system status as required.																												

SC_GEN14

SC_GEN14 is system common register 14.



Offset Address			Register Name																Total Reset Value													
0x00B8			SC_GEN14																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen14																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen14		This register is used to save the system status as required.																												

SC_GEN15

SC_GEN15 is system common register 15.

Offset Address			Register Name																Total Reset Value													
0x00BC			SC_GEN15																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen15																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen15		This register is used to save the system status as required.																												

SC_GEN16

SC_GEN16 is system common register 16.

Offset Address			Register Name																Total Reset Value													
0x00C0			SC_GEN16																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen16																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen16		This register is used to save the system status as required.																												

SC_GEN17

SC_GEN17 is system common register 17.



Offset Address			Register Name																Total Reset Value													
0x00C4			SC_GEN17																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen17																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen17		This register is used to save the system status as required.																												

SC_GEN18

SC_GEN18 is system common register 18.

Offset Address			Register Name																Total Reset Value													
0x00C8			SC_GEN18																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen18																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen18		This register is used to save the system status as required.																												

SC_GEN19

SC_GEN19 is system common register 19.

Offset Address			Register Name																Total Reset Value													
0x00CC			SC_GEN19																													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen19																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen19		This register is used to save the system status as required.																												

SC_GEN20

SC_GEN20 is system common register 20.



SC GEN21

SC_GEN21 is system common register 21.

Offset Address								Register Name								Total Reset Value																
0x00D4								SC_GEN21								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen21																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen21			This register is used to save the system status as required.																											

SC GEN22

SC_GEN22 is system common register 22.

SC GEN23

SC GEN23 is system common register 23.



Offset Address			Register Name																Total Reset Value													
0x00DC			SC_GEN23																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen23																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen23		This register is used to save the system status as required.																												

SC_GEN24

SC_GEN24 is system common register 24.

Offset Address			Register Name																Total Reset Value													
0x00E0			SC_GEN24																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen24																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen24		This register is used to save the system status as required. It is controlled by POR but not soft reset.																												

SC_GEN25

SC_GEN25 is system common register 25.

Offset Address			Register Name																Total Reset Value													
0x00E4			SC_GEN25																0x0000_0000													
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen25																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen25		This register is used to save the system status as required. It is controlled by POR but not soft reset.																												



SC_GEN26

SC_GEN26 is system common register 26.

Offset Address																Register Name								Total Reset Value								
0x00E8																SC_GEN26								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen26																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen26		This register is used to save the system status as required. It is controlled by POR but not soft reset.																												

SC_GEN27

SC_GEN27 is system common register 27.

Offset Address																Register Name								Total Reset Value								
0x00EC																SC_GEN27								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen27																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen27		This register is used to save the system status as required. It is controlled by POR but not soft reset.																												

SC_GEN28

SC_GEN28 is system common register 28.

Offset Address																Register Name								Total Reset Value								
0x00F0																SC_GEN28																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen28																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen28		This register is used to save the system status as required. It is controlled by POR but not soft reset.																												



SC_GEN29

SC_GEN29 is system common register 29.

Offset Address				Register Name												Total Reset Value																
0x00F4				SC_GEN29												0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen29																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen29			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC_GEN30

SC_GEN30 is system common register 30.

SC_GEN31

SC_GEN31 is system common register 31.

Offset Address								Register Name								Total Reset Value																
0x00FC								SC_GEN31								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen31																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen31			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											



SC_GPIO_OD_CTRL

SC_GPIO_OD_CTRL is a GPIO OD control register.

	Offset Address 0x0100																Register Name SC_GPIO_OD_CTRL								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	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		
Bits	Access	Name		Description																												
[31:7]	RW	reserved		Reserved																												
[6]	RW	gpio5_6_od_sel		Whether GPIO5 bit[6] is OD 0: no 1: yes																												
[5]	RW	gpio5_5_od_sel		Whether GPIO5 bit[5] is OD 0: no 1: yes																												
[4:3]	RW	reserved		Reserved																												
[2]	RW	gpio5_2_od_sel		Whether GPIO5 bit[2] is OD 0: no 1: yes																												
[1]	RW	gpio5_1_od_sel		Whether GPIO5 bit[1] is OD 0: no 1: yes																												
[0]	RW	gpio5_0_od_sel		Whether GPIO5 bit[0] is OD 0: no 1: yes																												

SC_LOCKEN

SC_LOCKEN is a write protection control register. This register provides the write protection mechanism for key registers (SC_CTRL and SC_SYSRES) of the system controller.



Offset Address																Register Name								Total Reset Value																								
Bit	0x020C															SC_LOCKEN								0x756E_4C4F																								
Name	sc_locken																																															
Reset	0 1 1 1 0 1 0 1 0 1 1 0 1 1 1 0 0 1 0 0 1 1 0 0 0 1 0 0 1 1 1 1																																															
Bits	Access	Name															Description																															
[31:0]	RW	sc_locken															Write protection indicator for key system registers. The configuration is as follows: When the value 0x4F50_454E is written to SC_LOCKEN, the write protection function of the preceding key registers is enabled. The attributes of the key registers are changed to writable, that is, these registers can be written. When any value except 0x4F50_454E is written to SC_LOCKEN, the attributes of the key register are changed to read-only. In this case, writing to these registers has no effect. You can check whether the key registers are write-protected by reading SC_LOCKEN. For example, if the value 0x756E_4C4F is returned after SC_LOCKEN is read, the key registers are not write-protected and writing to these registers has no effect. If the value 0x4C4F_434B is returned, write protection is enabled, and writing to these registers has no effect. The reset value of SC_LOCKEN is 0x756E_4C4F, indicating that the key registers are not write-protected.																															

SC_SYSID

SC_SYSID is a chip ID register.

Offset Address																Register Name								Total Reset Value																								
Bit	0x0EE0															SC_SYSID								0x3796_0100																								
Name	sc_sysid																																															
Reset	0 0 1 1 0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0																																															
Bits	Access	Name															Description																															
[31:0]	RO	sc_sysid															Major release of the chip																															

SC_GEN32

SC_GEN32 is system common register 32.



Offset Address								Register Name								Total Reset Value																
0x0F00								SC_GEN32								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen32			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC GEN33

SC_GEN33 is system common register 33.

Offset Address								Register Name								Total Reset Value																
0x0F04								SC_GEN33								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen33																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen33			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC GEN34

SC_GEN34 is system common register 34.

Offset Address								Register Name								Total Reset Value																
0x0F08								SC_GEN34								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen34																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen34			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											



SC_GEN35

SC_GEN35 is system common register 35.

Offset Address																Register Name								Total Reset Value								
0x0F0C																SC_GEN35								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen35																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen35			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC_GEN36

SC_GEN36 is system common register 36.

Offset Address																Register Name								Total Reset Value								
0x0F10																SC_GEN36								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen36																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen36			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC_GEN37

SC_GEN37 is system common register 37.

Offset Address																Register Name								Total Reset Value								
0x0F14																SC_GEN37																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen37																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen37			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											



SC_GEN38

SC_GEN38 is system common register 38.

Offset Address								Register Name								Total Reset Value																
0x0F18								SC_GEN38								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen38																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen38			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC_GEN39

SC_GEN39 is system common register 39.

SC_GEN40

SC_GEN40 is system common register 40.

Offset Address								Register Name								Total Reset Value																
0x0F20								SC_GEN40								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen40																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen40			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											



SC_GEN41

SC_GEN41 is system common register 41.

	Offset Address																Register Name																Total Reset Value															
	0x0F24																SC_GEN41																0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name	sc_gen41																																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen41				This register is used to save the system status as required. It is controlled by POR but not soft reset.																																										

SC_GEN42

SC_GEN42 is system common register 42.

	Offset Address																Register Name																Total Reset Value															
	0x0F28																SC_GEN42																0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name	sc_gen42																																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen42				This register is used to save the system status as required. It is controlled by POR but not soft reset.																																										

SC_GEN43

SC_GEN43 is system common register 43.

	Offset Address																Register Name																Total Reset Value															
	0x0F2C																SC_GEN43									0x0000_0000																						
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name	sc_gen43																																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen43				This register is used to save the system status as required. It is controlled by POR but not soft reset.																																										



SC_GEN44

SC_GEN44 is system common register 44.

Offset Address								Register Name								Total Reset Value																
0x0F30								SC_GEN44								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen44																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen44			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC GEN45

SC GEN45 is system common register 45.

Offset Address								Register Name								Total Reset Value																
0x0F34								SC_GEN45								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen45																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen45			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC GEN46

SC_GEN46 is system common register 46.

Offset Address								Register Name								Total Reset Value																
0x0F38								SC_GEN46								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen46																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen46			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											



SC_GEN47

SC_GEN47 is system common register 47.

	Offset Address																Register Name																Total Reset Value				
	0x0F3C																SC_GEN47																0x0000_0000				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	sc_gen47																																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen47	This register is used to save the system status as required. It is controlled by POR but not soft reset.																																		

SC_GEN48

SC_GEN48 is system common register 48.

	Offset Address																Register Name																Total Reset Value				
	0x0F40																SC_GEN48																0x0000_0000				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	sc_gen48																																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen48	This register is used to save the system status as required. It is controlled by POR but not soft reset.																																		

SC_GEN49

SC_GEN49 is system common register 49.

	Offset Address																Register Name																Total Reset Value				
	0x0F44																SC_GEN49																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name	sc_gen49																																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen49	This register is used to save the system status as required. It is controlled by POR but not soft reset.																																		



SC_GEN50

SC_GEN50 is system common register 50.

Offset Address			Register Name																Total Reset Value									
Bit	0x0F48			SC_GEN50																0x0000_0000								
Name				sc_gen50																								
Reset	0	0	0	0	0	0	0	0	0	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	sc_gen50			This register is used to save the system status as required. It is controlled by POR but not soft reset.																							

SC_GEN51

SC_GEN51 is system common register 51.

Offset Address			Register Name																Total Reset Value									
Bit	0x0F4C			SC_GEN51																0x0000_0000								
Name				sc_gen51																								
Reset	0	0	0	0	0	0	0	0	0	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	sc_gen51			This register is used to save the system status as required. It is controlled by POR but not soft reset.																							

SC_GEN52

SC_GEN52 is system common register 52.

Offset Address			Register Name																Total Reset Value									
Bit	0x0F50			SC_GEN52																0x0000_0000								
Name				sc_gen52																								
Reset	0	0	0	0	0	0	0	0	0	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	sc_gen52			This register is used to save the system status as required. It is controlled by POR but not soft reset.																							



SC_GEN53

SC_GEN53 is system common register 53.

Offset Address																Register Name								Total Reset Value								
0x0F54																SC_GEN53								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen53		This register is used to save the system status as required. It is controlled by POR but not soft reset.																												

SC_GEN54

SC_GEN54 is system common register 54.

Offset Address																Register Name								Total Reset Value								
0x0F58																SC_GEN54								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen54																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen54		This register is used to save the system status as required. It is controlled by POR but not soft reset.																												

SC_GEN55

SC_GEN55 is system common register 55.

Offset Address																Register Name								Total Reset Value								
0x0F5C																SC_GEN55																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen55																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen55		This register is used to save the system status as required. It is controlled by POR but not soft reset.																												



SC_GEN56

SC_GEN56 is system common register 56.

	Offset Address																Register Name																Total Reset Value															
	0x0F60																SC_GEN56																0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name	sc_gen56																																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen56														This register is used to save the system status as required. It is controlled by POR but not soft reset.																																

SC_GEN57

SC_GEN57 is system common register 57.

	Offset Address																Register Name																Total Reset Value															
	0x0F64																SC_GEN57																0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name	sc_gen57																																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen57														This register is used to save the system status as required. It is controlled by POR but not soft reset.																																

SC_GEN58

SC_GEN58 is system common register 58.

	Offset Address																Register Name																Total Reset Value															
	0x0F68																SC_GEN58																0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name	sc_gen58																																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen58														This register is used to save the system status as required. It is controlled by POR but not soft reset.																																



SC_GEN59

SC_GEN59 is system common register 59.

Offset Address								Register Name								Total Reset Value																
0x0F6C								SC_GEN59								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen59																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen59			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC_GEN60

SC_GEN60 is system common register 60.

SC_GEN61

SC_GEN61 is system common register 61.

Offset Address								Register Name								Total Reset Value																
0x0F74								SC_GEN61								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen61																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen61			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											



SC_GEN62

SC_GEN62 is system common register 62.

Offset Address								Register Name								Total Reset Value																
0x0F78								SC_GEN62								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	sc_gen62																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	sc_gen62			This register is used to save the system status as required. It is controlled by POR but not soft reset.																											

SC_GEN63

SC_GEN63 is system common register 63.

3.5 Peripheral Controller

3.5.1 Overview

The peripheral controller controls peripherals, queries their status, and configures some functions of peripherals.

3.5.2 Features

The peripheral controller has the following features:

- Controls read and write operations on peripherals over the advanced peripheral bus (APB).
 - Controls some peripherals and queries their status.
 - Controls the software interrupts between CPUs.



3.5.3 Register Summary

Table 3-45 describes PERI_CTRL registers.

Table 3-45 Summary of PERI_CTRL registers (base address: 0xF8A2_0000)

Offset Address	Register	Description	Page
0x0000	START_MODE	System startup status query register	3-109
0x0008	PERI_CTRL	Peripheral control register	3-110
0x000C	CPU_STAT	A7 status query register	3-112
0x00B4	PERI_USB_RESUME_INT_MASK	USB wakeup interrupt mask register	3-112
0x00B8	PERI_USB_RESUME_INT_RAWSTAT	Raw USB wakeup interrupt status register	3-113
0x00BC	PERI_USB_RESUME_INT_STAT	USB wakeup interrupt status register	3-114
0x00C0	PERI_INT_A9TOMCE	A7-to-MCU software interrupt register	3-114
0x00E4	PERI_INT_SWI0	Software interrupt 0 register	3-115
0x00E8	PERI_INT_SWI1	Software interrupt 1 register	3-115
0x00EC	PERI_INT_SWI2	Software interrupt 2 register	3-116
0x00F0	PERI_INT_SWI0_MASK	Software interrupt 0 mask register	3-116
0x00F4	PERI_INT_SWI1_MASK	Software interrupt 1 mask register	3-117
0x00F8	PERI_INT_SWI2_MASK	Software interrupt 2 mask register	3-118
0x0110	PERI_TIANLA_ADAC0	ADAC configuration register 0	3-118
0x0114	PERI_TIANLA_ADAC1	ADAC configuration register 1	3-120
0x0118	PERI_FEPHY	FE PHY configuration register	3-123
0x011C	PERI_SD_LDO	SDIO0 LDO control register	3-124
0x0120	PERI_USB0	USB controller configuration register 0	3-125
0x0124	PERI_USB1	USB PHY configuration register	3-127
0x0134	PERI_USB5	USB 3.0 system controller 0	3-127
0x0138	PERI_USB6	USB 3.0 physical coding sublayer (PCS) system controller 1	3-129
0x013C	PERI_USB7	USB 3.0 PHY system controller 0	3-129
0x0140	PERI_USB8	USB 3.0 PHY system controller 1	3-130



Offset Address	Register	Description	Page
0x0144	PERI_USB9	USB 3.0 PHY system controller 2	3-132
0x0148	PERI_USB10	USB 3.0 PHY system controller 3	3-133
0x0150	PERI_USB12	USB 3.0 system controller 2	3-134
0x0154	PERI_USB13	USB controller configuration register 1	3-135
0x0158	PERI_USB14	USB PHY configuration register 1	3-136
0x01E0	CHIPSET_INFO	Chip information register	3-137
0x01F0	PERI_SW_SET	Software flag configuration register	3-137
0x083C	PERI_SIM_OD_CTRL	SCI pin output OD mode control register	3-137
0x0844	PERI_FEPHY_LDO_CTRL	FE PHY LDO control register	3-138
0x880	PERI_CPU_A7_CFG	CPU control register	3-138
0x884	PERI_CPU_A7_STAT	CPU status register	3-140
0x0890	PERI_FE_OD_CTRL	FE LED pin output OD mode control register	3-141

3.5.4 Register Description

START_MODE

START_MODE is a system startup status query register.

	Offset Address		Register Name		Total Reset Value																											
	0x0000		START_MODE		0x0140_0700																											
Bit	31	30	29	28	27	26	25	24	23	22	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_boot_jtag_sel_in	reserved	reserved		boot_sel	reserved																								
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0		
Bits	Access	Name		Description																												
[31:25]	RO	reserved		Reserved																												



[24]	RO	usb_boot	USB device burning boot enable 0: enabled 1: disabled
[23]	RO	jtag_sel_in	Multiplexing control indicator of JTAG pins 0: JTAG pin multiplexing is controlled by pin multiplexing registers. For detail, see the pin multiplexing section. 1: The JTAG pins are always used for JTAG functions. JTAG pins include JTAG_TDI, JTAG_TCK, JTAG_TMS, JTAG_TDO, and JTAG_TRSTN.
[22:11]	RO	reserved	Reserved
[10:9]	RO	boot_sel	Boot memory type 00: reserved 01: NAND flash 10: SD 11: eMMC
[8:0]	RO	reserved	Reserved

PERI_CTRL

PERI_CTRL is a peripheral control register.

	Offset Address																Register Name								Total Reset Value							
	0x0008																PERI_CTRL								0x0400_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	peri_jtag_sel		clk_mhlrx_sel	reserved	usb3_ip_iso_ctrl	reserved																ssp0_cs_sel	usb2_ip_iso_ctrl	reserved	sdio0_card_det_mode	sdio1_card_det_mode						
Reset	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																									



[31:29]	RW	peri_jtag_sel	JTAG interface select when jtag_sel is 0 0: CPU JTAG 1: reserved 2: reserved 3: reserved 4: reserved 5: reserved 6: reserved 7: SOC JTAG
[28]	RW	clk_mhlnx_sel	HDMI MHLNX clock select 0: reduced clock path (this bit must be 0 when the frequency is 268.5 MHz) 1: original clock path
[27]	RW	reserved	Reserved
[26]	RW	usb3_ip_iso_ctrl	USB3 PHY IP isolation control 0: not isolated 1: isolated
[25:5]	RW	reserved	Reserved
[4]	RW	ssp0_cs_sel	SSP0 CS multiplexing control 0: SSP0 CS0 1: SSP0 CS1
[3]	RW	usb2_ip_iso_ctrl	USB2_1P IP isolation control 0: not isolated 1: isolated
[2]	RW	reserved	Reserved
[1]	RW	sdio0_card_det_mode	Polarity of the SDIO0 card detection signal 0: active low 1: active high
[0]	RW	sdio1_card_det_mode	Polarity of the SDIO1 card detection signal 0: active low 1: active high



CPU_STAT

CPU_STAT is a CPU status query register.

	Offset Address																Register Name																Total Reset Value							
	0x000C																CPU_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																									smpnamp														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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																						
[3:0]	RO	smpnamp																A7MP core SMP or AMP indicator (bit 3–bit 0 correspond to core 3–core 0 respectively) 0: SMP mode 1: AMP mode																						

PERI_USB_RESUME_INT_MASK

PERI_USB_RESUME_INT_MASK is a USB wakeup interrupt mask register.

	Offset Address																Register Name																Total Reset Value							
	0x00B4																PERI_USB_RESUME_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																										usb_phy1_suspend_int_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	0	0	0	0	0			
Bits	Access	Name																Description																						
[31:4]	RW	reserved																Reserved																						
[3]	RW	usb_phy1_suspend_int_mask																USB PHY1 wakeup interrupt mask 0: masked 1: not masked																usb_phy2_suspend_int_mask						



[2]	RW	usb_phy2_suspend_int_mask	USB PHY2 wakeup interrupt mask 0: masked 1: not masked
[1]	RW	usb3_utmi_suspend_n	USB3 UTMI wakeup interrupt mask 0: masked 1: not masked
[0]	RW	usb_phy0_suspend_int_mask	USB PHY0 wakeup interrupt mask 0: masked 1: not masked

PERI_USB_RESUME_INT_RAWSTAT

PERI_USB_RESUME_INT_RAWSTAT is a raw USB wakeup interrupt status register.

	Offset Address 0x00B8 Register Name PERI_USB_RESUME_INT_RAWSTAT																Total Reset Value 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																																
Reset	0	0	0	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]	RO	reserved			Reserved																												
[3]	RO	usb_phy1_suspend_int_rawstat			USB PHY1 suspend interrupt raw status 0: No wakeup interrupt is generated. 1: A wakeup interrupt is generated.																											usb_phy1_suspend_int_rawstat	
[2]	RO	usb_phy2_suspend_int_rawstat			USB PHY2 suspend interrupt raw status 0: No wakeup interrupt is generated. 1: A wakeup interrupt is generated.																										usb_phy2_suspend_int_rawstat		
[1]	RO	usb3_suspend_int_rawstat			USB3 UTMI suspend interrupt raw status 0: No wakeup interrupt is generated. 1: A wakeup interrupt is generated.																										usb3_suspend_int_rawstat		
																																usb3_suspend_int_rawstat	



[0]	RO	usb_phy0_suspend_int_rawstat	USB PHY0 suspend interrupt raw status 0: No wakeup interrupt is generated. 1: A wakeup interrupt is generated.
-----	----	------------------------------	--

PERI_USB_RESUME_INT_STAT

PERI_USB_RESUME_INT_STAT is a USB wakeup interrupt status register.

	Offset Address		Register Name	Total Reset Value																
Bit	0x00BC PERI_USB_RESUME_INT_STAT																0x0000_0000			
Name	reserved																usb_phy1_suspend_int_stat	usb_phy2_suspend_int_stat	usb3_suspend_int_stat	usb_phy0_suspend_int_stat
Reset	0	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]	RO	usb_phy1_suspend_int_stat	USB PHY1 suspend interrupt status 0: No wakeup interrupt is generated. 1: A wakeup interrupt is generated.																	
[2]	RO	usb_phy2_suspend_int_stat	USB PHY2 suspend interrupt status 0: No wakeup interrupt is generated. 1: A wakeup interrupt is generated.																	
[1]	RO	usb3_suspend_int_stat	USB3 UTMI suspend interrupt status 0: No wakeup interrupt is generated. 1: A wakeup interrupt is generated.																	
[0]	RO	usb_phy0_suspend_int_stat	USB PHY0 suspend interrupt status 0: No wakeup interrupt is generated. 1: A wakeup interrupt is generated.																	

PERI_INT_A9TOMCE

PERI_INT_A9TOMCE is a CPU-to-MCU software interrupt register.



Offset Address				Register Name												Total Reset Value																
0x00C0				PERI_INT_A9TOMCE												0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	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_a9tomce		
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RW		reserved				Reserved																									
[0]	RW		int_a9tomce				CPU-to-MCU interrupt 0: clear the interrupt 1: send an interrupt request																									

PERI INT SWI0

PERI INT SWI0 is a software interrupt 0 register.

PERI_INT_SWI1

PERI INT SWI1 is a software interrupt 1 register.



PERI INT SWI2

PERI_INT_SWI2 is a software interrupt 2 register.

PERI_INT_SWI0_MASK

PERI_INT_SWI0_MASK is a software interrupt 0 mask register.



	Offset Address	Register Name	Total Reset Value
	0x00F0	PERI_INT_SWI0_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	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 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]	RW	reserved	Reserved
[0]	RW	int_swi0_mask_a9	Mask control for software interrupt 0 transmitted to the CPU 0: masked 1: enabled

PERI_INT_SWI1_MASK

PERI_INT_SWI1_MASK is a software interrupt 1 mask register.

	Offset Address	Register Name	Total Reset Value
	0x00F4	PERI_INT_SWI1_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	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 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]	RW	reserved	Reserved
[0]	RW	int_swi1_mask_a9	Mask control for software interrupt 1 transmitted to the CPU 0: masked 1: enabled



PERI_INT_SWI2_MASK

PERI_INT_SWI2_MASK is a software interrupt 2 mask register.

PERI TIANLA ADAC0

PERI TIANLA ADAC0 is ADAC configuration register 0.

Offset Address								Register Name								Total Reset Value																
0x0110								PERI_TIANLA_ADAC0								0xF840_0606																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	pd_dacl	pd_dacr	mute_dacl	mute_dacr	pd_vref	fs	popfref	popfref	dac1_path	dac1_path	dac1_deemph	dac1_deemph	deemphasis_fs		reserved		dacl_vol				reserved		dacr_vol									
Reset	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0		
Bits		Access		Name				Description																								
[31]		RW		pd_dacl				Power-down control for the DAC audio-left channel 0: The digital and analog circuits work properly. 1: The digital circuit works properly, whereas the analog circuit is powered down.																								



[30]	RW	pd_daer	Power-down control for the DAC audio-right channel 0: The digital and analog circuits work properly. 1: The digital circuit works properly, whereas the analog circuit is powered down.
[29]	RW	mute_dacl	Mute control for the DAC audio-left channel 0: The digital and analog channels work properly. 1: The digital and analog audio-left channels are muted.
[28]	RW	mute_dacr	Mute control for the DAC audio-right channel 0: The digital and analog audio-right channels work properly. 1: The digital and analog audio-right channels are muted.
[27]	RW	pd_vref	Power-down control for the reference voltage 0: powered on 1: powered down
[26]	RW	fs	Fast startup enable for the reference voltage 0: disabled 1: enabled
[25]	RW	popfreel	Pop-free enable for the DAC audio-left channel 0: disabled 1: enabled
[24]	RW	popfreer	Pop-free enable for the DAC audio-right channel 0: disabled 1: enabled
[23]	RW	dACL_path	Data source of the DAC audio-left channel 0: RX data 1 from the I ² S interface (audio-left channel) 1: RX data 2 from the I ² S interface (audio-right channel)
[22]	RW	dACR_path	Data source of the DAC audio-right channel 0: RX data 1 from the I ² S interface (audio-left channel) 1: RX data 2 from the I ² S interface (audio-right channel)
[21]	RW	dACL_deemph	De-emphasis filter enable for the DAC audio-left channel 1: enabled 0: disabled
[20]	RW	dACR_deemph	De-emphasis filter enable for the DAC audio-right channel 1: enabled 0: disabled



[19:18]	RW	deemphasis_fs	De-emphasis sampling rate 00: 48 kHz 01: 44.1 kHz 10: 32 kHz 11: reserved
[17:15]	RW	reserved	Reserved
[14:8]	RW	dacl_vol	Output volume of the audio-left channel 0x00: 6 dB 0x01: 5 dB 0x06: 0 dB ... (The step is 1 dB.) 0x7F: -121 dB
[7]	RW	reserved	Reserved
[6:0]	RW	dacr_vol	Output volume of the audio-right channel 0x00: 6 dB 0x01: 5 dB 0x06: 0 dB ... (The step is 1 dB.) 0x7F: -121 dB

PERI_TIANLA_ADAC1

PERI_TIANLA_ADAC1 is ADAC configuration register 1.

Offset Address								Register Name								Total Reset Value																
0x0114								PERI_TIANLA_ADAC1								0x00E0_6755																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	smutel	smuter	summute1	summute2	dacv1	mute_rate	reserved	data_bits	sample_sel	reserved	reserved	reserved	adj_dac	adj_ctem	rst	adj_refbf	clkse12	clkdgesel	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved		
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	1	1	1	0	0	1	0	1	0	1		
Bits		Access		Name				Description																								
[31]	RW	smutel	Soft mute control for the audio-left channel																0: disabled				1: enabled									



[30]	RW	smuter	Soft mute control for the audio-right channel 0: disabled 1: enabled
[29]	RW	sunmutel	Soft unmute control for the audio-left channel of the digital circuit 0: disabled 1: enabled
[28]	RW	sunmuter	Soft unmute control for the audio-right channel of the digital circuit 0: disabled 1: enabled
[27]	RW	dacvu	Volume sync update of the digital audio-right and audio-left channels 0: The gains of the audio-left and audio-right channels remain unchanged. 1: The volumes of audio-left and audio-right channels are simultaneously updated.
[26:25]	RW	mute_rate	Volume gain step when the audio-right and audio-left channels are soft muted or soft unmuted 00: fs/2 01: fs/8 10: fs/32 11: fs/64
[24]	RW	reserved	Reserved
[23:22]	RW	data_bits	Interface data bit width 00: 16 bits 01: 18 bits 10: 20 bits 11: 24 bits
[21:19]	RW	sample_sel	Sampling rate 000: 1/4 of the reference sampling rate 001: 1/2 of the reference sampling rate 010: reference sampling rate 011: 2 times the reference sampling rate 100: 4 times the reference sampling rate Other values: reference sampling rate
[18:16]	RW	reserved	Reserved



[15:14]	RW	adj_dac	DAC bias current 00: 0.6 times the current 01: normal current (recommended) 10: 1.4 times the current 11: 2 times the current
[13]	RW	adj_ctcm	CTCM bias current 0: normal current (recommended) 1: 2 times the current
[12]	RW	rst	STB_DAC_ANA reset 0: normal mode 1: reset
[11:10]	RW	adj_refbf	Reference buffer bias current 00: 0.6 times the current 01: normal current (recommended) 10: 1.4 times the current 11: 2 times the current
[9]	RW	clksel2	2-divider clock select 0: divide-by-2 clock 1: input clock
[8]	RW	clkdgesel	Clock edge select 0: The rising edge is selected as the sampling clock of the analog part. 1: The falling edge is selected as the sampling clock of the analog part.
[7:0]	RW	reserved	Reserved



PERI_FEPHY

PERI_FEPHY is an FE PHY configuration register.

Offset Address										Register Name								Total Reset Value														
0x0118										PERI_FEPHY								0x0300_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	soft_fephy_gp_i										reserved	soft_fephy_mdio_i	reserved	soft_fephy_mdio_mdc	reserved	fephy_patch_enable	reserved	fephy_tclk_enable	reserved	reserved	fephy_phy_addr											
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	1			
Bits	Access	Name		Description																												
[31:16]	RW	soft_fephy_gp_i		GP data input when the FE PHY downloads patches																												
[15]	RW	reserved		Reserved																												
[14]	RW	soft_fephy_mdio_i		MDIO data input when the FE PHY downloads patches																												
[13]	RW	reserved		Reserved																												
[12]	RW	soft_fephy_mdio_mdc		MDC data input when the FE PHY downloads patches																												
[11]	RW	reserved		Reserved																												
[10]	RW	fephy_patch_enable		FE PHY download patch enable 0: disabled 1: enabled																												
[9]	RW	reserved		Reserved																												
[8]	RW	fephy_tclk_enable		FE PHY debugging enable. 0: disabled 1: enabled																												
[7:5]	RW	reserved		Reserved																												
[4:0]	RW	fephy_phy_addr		FE PHY address																												



PERI_SD_LDO

PERI_SD_LDO is an SDIO0 LDO control register.

Offset Address								Register Name								Total Reset Value																
0x011C								PERI_SD_LDO								0x0000_0060																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																bypass	en	vset	fuse												
Reset	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	0	0	0		
Bits	Access	Name		Description																												
[31:7]	RO	reserved		Reserved																												
[6]	RW	bypass		LDO bypass enable 0: disabled 1: enabled (the VIN voltage is directly output)																												
[5]	RW	en		LDO enable 0: disabled 1: enabled																												
[4]	RW	vset		LDO output voltage 0: 1.8 V 1: 1.2 V																												
[3:0]	RW	fuse		Reference voltage for accurately adjusting the internal voltage																												



PERI_USB0

PERI_USB0 is USB 2.0 controller configuration register 0.

Bit	Offset Address 0x0120																Register Name PERI_USB0								Total Reset Value 0x000C_71A8									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	ohci_0_cntsel_i_n	ss_scaledown_mode	chipid	reserved				ss_hubsetup_min_i	reserved												ss_ena_incr16_i	ss_ena_incr8_i	ss_ena_incr4_i	ss_ena_incr_align_i	ss_autopd_on_overcur_en_i	reserved	ulp_bypass_en_i	app_start_clk_i	ohci_susp_lgey_i	ss_word_if_i				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	1	1	0	1	0	0	0	0			
Bits	Access	Name		Description																														
[31]	RW	ohci_0_cntsel_i_n		Interval for transmitting USB SOF packets 0: SOF packets are transmitted every 1 ms (normal interval). 1: The interval is reduced for simulation.																														
[30:29]	RW	ss_scaledown_mod_e		Scale-down mode 00: Scale-down is disabled for all timings, and the actual timings are used. 01: Scale-down is enabled for all timings except the suspended and resumed timings in device mode. 10: Scale-down is enabled only for the suspended and resumed timings in device mode. 11: Scale-down timings of bit 0 and bit 1 are enabled.																														
[28]	RW	chipid		USB PHY controller select 0: USB host port 0 1: reserved																														
[27:23]	RW	reserved		Reserved																														
[22]	RW	ss_hubsetup_min_i		Number of idle cycles after the full-speed preamble packet 0: 5 full-speed idle cycles 1: 4 full-speed idle cycles																														
[21:10]	RW	reserved		Reserved																														
[9]	RW	ss_ena_incr16_i		AHB burst 16 enable 0: disabled 1: enabled (default)																														



[8]	RW	ss_ena_incr8_i	AHB burst 8 enable 0: disabled 1: enabled (default)
[7]	RW	ss_ena_incr4_i	AHB burst 4 enable 0: disabled 1: enabled (default)
[6]	RW	ss_ena_incr_align_i	Burst alignment enable 0: disabled (default) 1: enabled
[5]	RW	ss_autoppd_on_overcur_en_i	Automatic port power shutdown enable during overcurrent 0: disabled 1: enabled (default)
[4]	-	reserved	Reserved
[3]	RW	ulpi_bypass_en_i	UTMI+ low pin interface (ULPI) bypass control. This bit must be set to 1. 0: ULPI mode 1: UTMI mode
[2]	RW	app_start_clk_i	Open host controller interface (OHCI) clock control signal. 0: The OHCI works properly. (default) 1: The OHCI clock is enabled in suspend mode.
[1]	RW	ohci_susp_lgcy_i	Strap input signal when the OHCI is suspended
[0]	RW	ss_word_if_i	Data bit width select signal of the UTMI interface 0: 8 bits (default) 1: 16 bits



PERI_USB1

PERI_USB1 is a USB 2.0 PHY configuration register.

	Offset Address								Register Name								Total Reset Value															
	0x0124								PERI_USB1								0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	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_rddata								test_rstn	test_clk	test_wren	test_addr								test_wrdata				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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]	RO	test_rddata		Data read signal of the debugging channel																												
[15]	RW	test_rstn		Reset signal of the debugging channel																												
[14]	RW	test_clk		Clock signal of the debugging channel																												
[13]	RW	test_wren		Read/Write enable signal of the debugging channel																												
[12:8]	RW	test_addr		Address signal of the debugging channel																												
[7:0]	RW	test_wrdata		Data write signal of the debugging channel																												

PERI_USB5

PERI_USB5 is USB 3.0 control register 0.

	Offset Address								Register Name								Total Reset Value																	
	0x0134								PERI_USB5								0x7E8F_2040																	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	host_current_belt								bus_filter_bypass				reserved		fladj_30mhz_reg								host_msi_enable		host_port_power_control_present		hub_u3_port_disable		hub_u2_port_disable		hub_port_perm_attach		reserved	
Reset	0	1	1	1	1	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0		



Bits	Access	Name	Description
[31:20]	RO	host_current_belt	Current BELT value
[19:16]	RW	bus_filter_bypass	UTMI bus signal filter enable bus_filter_bypass[3] The function of filtering out the utmiotg_iddig signal is disabled. bus_filter_bypass[2] The function of filtering out the utmisrp_bvalid and utmisrp_sessend signals is disabled. bus_filter_bypass[1] The function of filtering out the pipe3_PowerPresent signal of port U3 is disabled. bus_filter_bypass[0] The function of filtering out the utmiotg_vbusvalid signal of port U2 is disabled.
[15:14]	RW	reserved	Reserved
[13:8]	RW	fladj_30mhz_reg	High-speed signal jitter adjustment mac3_clock and utmi_clock are adjusted to 125 μs.
[7]	RW	host_msi_enable	Pulse interrupt enable
[6]	RW	host_port_power_c ontrol_present	Port power switch enable 1'b0: The port has no power switch. 1'b1: The port has a power switch.
[5]	RW	hub_u3_port_disabl e	USB 3.0 super-speed port enable 1'b0: enabled 1'b1: disabled
[4]	RW	hub_u2_port_disabl e	USB 3.0 high-speed port enable 1'b0: The port is enabled. 1'b1: The port is disabled.
[3:2]	RW	hub_port_perm_att ach	Device permanent insertion 1'b0: The device is inserted permanently. 1'b1: The device is not inserted permanently.
[1:0]	RW	reserved	Reserved



PERI_USB6

PERI_USB6 is USB 3.0 PCS control register 1.

	Offset Address								Register Name								Total Reset Value															
	0x0138								PERI_USB6								0x5D81_560D															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	pcs_tx_swing_full								pcs_tx_deemph_6db	pcs_tx_deemph_3p5db								lane0_tx2rx_lopbk	lane0_power_present_ovrd_en	lane0_power_present_ovrd	lane0_ext_pclk_req	reserved	usb_pwr_ctrl								
Reset	0	1	0	1	1	1	0	1	1	0	0	0	0	0	0	1	0	1	0	1	0	1	1	0	0	0	0	1	1	0	1	
Bits	Access																															
[31]	RW	reserved		Reserved																												
[30:24]	RW	pcs_tx_swing_full		TX signal swing (for the eye pattern test)																												
[23:18]	RW	pcs_tx_deemph_6db		Preemphasis of 6 dB reduction (for the eye pattern test)																												
[17:12]	RW	pcs_tx_deemph_3p5db		Preemphasis of 3.5 dB reduction (for the eye pattern test)																												
[11]	RW	lane0_tx2rx_lopbk		Lane 0 TX-to-RX loopback enable																												
[10]	RW	lane0_power_present_ovrd_en		Lane 0 overcurrent protection enable																												
[9]	RW	lane0_power_present_ovrd		Lane 0 overcurrent protection																												
[8]	RW	lane0_ext_pclk_req		Lane 0 external pipe clock enable																												
[7:4]	RW	reserved		Reserved																												
[3:0]	RW	usb_pwr_ctrl		Power control usb_pwr_ctrl[0]: power switch usb_pwr_ctrl[1]: overcurrent protection enable usb_pwr_ctrl[2]: power switch polarity usb_pwr_ctrl[3]: power switch polarity																												

PERI_USB7

PERI_USB7 is USB 3.0 PHY system controller 0.



Offset Address																Register Name								Total Reset Value													
Bit	0x013C															PERI_USB7							0x0000_0000														
Name	crdatain															crdataout																					
Reset	0 0																																				
Bits	Access	Name															Description																				
[31:16]	RO	crdatain															Current input data																				
[15:0]	RW	crdataout															Current output data																				

PERI_USB8

PERI_USB8 is USB 3.0 PHY system controller 1.

Offset Address																Register Name								Total Reset Value							
Bit	0x0140															PERI_USB8							0x9689_A100								
Name	otgtune	otgdisable0	mpllrefssclken	mpllmultiplier												loslevel	losbias	loopbackenb0	idpullup0	iddig0	drvibus0	compdistune0	commonon0	crack	crwrite	erread	creaddata	crecapaddr			
Reset	1 0 0 1	0 1 1 0	1 0 0 0 0	1 0 0 1	1 0 1 0	1 0 0 0 0	1 0 0 1	1 0 1 0	1 0 0 1	0 0 0 0 0 0	1 0 0 1	1 0 0 1	0 0 0 0 0 0 0 0	1 0 0 1	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 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 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]	RW	otgtune															Vbus valid threshold adjust 1 bit = 1.5%														
[28]	RW	otgdisable0															PHY OTG disable														
[27]	RW	mpllrefssclken															Spread reference clock output. It is 20 MHz and provides the clock for the hardcore of the clock source of ref_alt_clk.														
[26:20]	RW	mpllmultiplier															MPLL frequency multiplier control for obtaining a specified working frequency														
[19:16]	RW	loslevel															Sensitivity level detection for the loss-of-signal (LOS) detector														
[15:13]	RW	losbias															Level threshold detection for the LOS detector														
[12]	RW	loopbackenb0															Loopback enable														
[11]	RW	idpullup0															Sampling enable for the analog ID input signal														



[10]	RO	iddig0	Mini-A or mini-B connector connected to the PHY 1: mini-B connector 0: mini-A connector
[9]	RW	drvbus0	Vbus valid comparator enable 1: enabled 0: disabled
[8:6]	RW	compdistune0	Disconnection event detection voltage threshold. This field is used to set the voltage threshold for the disconnection event between the device and the host.
[5]	RW	commomon0	Common module enable 1: The HS bias and PLL modules are disabled in suspend or sleep mode. 0: The HS bias and PLL modules are enabled in suspend or sleep mode.
[4]	RO	crack	Controller signal (as the response to the CRWRITE, CRREAD, CRCAPDATA, and CRCAPSDDR signals)
[3]	RW	crwrite	Register write control signal
[2]	RW	crread	Register read control signal
[1]	RW	crcapdata	Register data capture control signal. cr_data_in[15:0] are transferred to the written data.
[0]	RW	crcapaddr	Register address capture control signal. cr_data_in[15:0] are transferred to the address register.



PERI_USB9

PERI_USB9 is USB 3.0 PHY system controller 2.

Offset Address												Register Name												Total Reset Value											
0x0144												PERI_USB9												0x0684_6017											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	txpreemppulse0	txpreempamp tune0	txfuslstune0	sscrefclk sel	reserved	txhsxvtune0	tx0termoffset	ssrange	sscen	sqrxtune0	retenable																								
Reset	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	1					
Bits	Access	Name		Description																															
[31]	RW	txpreemppulse0		TX end preemphasis time in high-speed mode 1: 580 μ s 0: 2x580 μ s																															
[30:29]	RW	txpreempamp tune0		TX end preemphasis tune in high-speed mode 11: 1800 μ A 10: 1200 μ A 01: 600 μ A 00: Preemphasis is disabled																															
[28:25]	RW	txfuslstune0		Source impedance tune in full-speed or low-speed mode Hot codes are used. The impedance is reduced by 2.5% each time a hot code is added, and increased by 2.5% each time a hot code is reduced.																															
[24:16]	RW	sscrefclk sel		Spread spectrum reference clock select																															
[15]	RW	reserved		Reserved																															
[14:13]	RW	txhsxvtune0		TX end DP/DM voltage tune in high-speed mode 11: default configuration 10: +15 mV 01: -15 mV 00: reserved																															
[12:8]	RW	tx0termoffset		TX termination compensation tune enable																															
[7:5]	RW	ssrange		Spread spectrum clock range																															
[4]	RW	sscen		Spread spectrum enable																															



[3:1]	RW	sqrxtune0	High-speed data detection level tune The level is decreased by 5% each time a binary value is added, and increased by 5% each time a binary value is reduced.
[0]	RW	retenablen	Low digital power indicator, indicating that the VP digital power is decreased in suspend mode 1: The normal operating mode is used. 0: The analog power is shut down.

PERI_USB10

PERI_USB10 is USB 3.0 PHY system controller 3.

	Offset Address 0x0148																Register Name PERI_USB10																Total Reset Value 0x0006_7005															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Name	reserved																vbusvldexsel0	vbusvldex0	txvrectune0	txboostlvl	testpowerdownssp	testpowerdownhsp	rtunek	rtunreq	vdaisrcenb0	vdatdetenb0	txrisetune0	txrestune0																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1																
Bits	Access	Name	Description																																													
[31:19]	RW	reserved	Reserved																																													
[18]	RW	vbusvldexsel0	External Vbus valid indicator signal 1: VBUSVLDEXT0 0: internal comparator																																													
[17]	RW	vbusvldex0	External Vbus valid indicator signal 1: The Vbus0 signal and the pull-up resistor on D+ are valid. 0: The Vbus0 signal is invalid, and the pull-up resistor on D+ is disabled.																																													
[16:13]	RW	txvrectune0	DC voltage tune in high-speed mode The voltage is increased by 1.25% each time the binary value 1 is added, and decreased by 1.25% each time the binary value 1 is reduced.																																													
[12:10]	RW	txboostlvl	TX voltage increase																																													
[9]	RW	testpowerdownssp	Power-off control for the super-speed functional circuit																																													
[8]	RW	testpowerdownhsp	Power-off control for the high-speed functional circuit																																													



[7]	RO	rtunreq	Impedance tune acknowledge enable
[6]	RW	rtunreq	Impedance tune request
[5]	RW	vdatsrcenb0	Charging power select 1: The data source voltage (VDAT_SRC) is enabled. 0: The data source voltage (VDAT_SRC) is disabled.
[4]	RW	vdatdetenb0	Charging connection/disconnection detection enable 1: The data detection voltage is enabled. 0: The data detection voltage is disabled.
[3:2]	RW	txrisetune0	Rising/Falling edge time tune for the TX end in high-speed mode The rising/falling edge time is decreased by 4% each time the binary value 1 is added, and increased by 4% each time the binary value 1 is reduced.
[1:0]	RW	txrestune0	USB matched source impedance tune

PERI_USB12

PERI_USB12 is USB 3.0 system controller 2.

	Offset Address																Register Name																Total Reset Value															
	0x0150																PERI_USB12																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																									pcs_rx_los_mask_val																						
Reset	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																																											
[31:10]	RW	reserved			Reserved																																											
[9:0]	RW	pcs_rx_los_mask_val			Number of reference clock cycles to mask the incoming LFPS For normal operations, set this bit to 10 µs/Tref_clk (a targeted mask interval of 10 µs). If the ref_clkdiv2 signal is used, set this bit to 10 µs/(2 x Tref_clk).																																											



PERI_USB13

PERI_USB13 is USB controller configuration register 1.

	Offset Address	Register Name	Total Reset Value
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	PERI_USB13	0x000C_71A8
Name	ohci_0_cntsel_i_n reserved ss_hubsetup_min_i reserved	ss_ena_incr16_i ss_ena_incr8_i ss_ena_incr4_i ss_ena_incr_align_i ss_autopd_on_overcur_en_i reserved ulp_bypass_en_i app_start_clk_i ohci_susp_lgey_i ss_word_if_i	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 0 0 0 1 1 0 1 0 1 0 0 0		
Bits	Access	Name	Description
[31]	RW	ohci_0_cntsel_i_n	Interval for transmitting USB SOF packets 0: SOF packets are transmitted every 1 ms (normal interval). 1: The interval is reduced for simulation.
[30:23]	RW	reserved	Reserved
[22]	RW	ss_hubsetup_min_i	Number of idle cycles after the full-speed preamble packet 0: 5 full-speed idle cycles 1: 4 full-speed idle cycles
[21:10]	RW	reserved	Reserved
[9]	RW	ss_ena_incr16_i	AHB burst 16 enable 0: disabled 1: enabled (default)
[8]	RW	ss_ena_incr8_i	AHB burst 8 enable 0: disabled 1: enabled (default)
[7]	RW	ss_ena_incr4_i	AHB burst 4 enable 0: disabled 1: enabled (default)
[6]	RW	ss_ena_incr_align_i	Burst alignment enable 0: disabled (default) 1: enabled



[5]	RW	ss_autoppd_on_overcur_en_i	Automatic port power shutdown enable during overcurrent 0: disabled 1: enabled (default)
[4]	RW	reserved	Reserved
[3]	RW	ulpqi_bypass_en_i	ULPI bypass control. This bit must be set to 1. 0: ULPI mode 1: UTMI mode
[2]	RW	app_start_clk_i	OHCI clock control signal 0: The OHCI works properly. (default) 1: The OHCI clock is enabled in suspend mode.
[1]	RW	ohci_susp_lgcy_i	Strap input signal when the OHCI is suspended. The default value is 0.
[0]	RW	ss_word_if_i	Data bit width select signal of the UTMI interface 0: 8 bits (default) 1: 16 bits

PERI_USB14

PERI_USB14 is USB PHY configuration register 1.

	Offset Address 0x0158																Register Name PERI_USB14								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	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_rddata								test_rstn	test_clk	test_wren	test_addr				test_wrdata								
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RO	test_rddata				Data read signal of the debugging channel																										
[15]	RW	test_rstn				Reset signal of the debugging channel																										
[14]	RW	test_clk				Clock signal of the debugging channel																										
[13]	RW	test_wren				Read/Write enable signal of the debugging channel																										
[12:8]	RW	test_addr				Address signal of the debugging channel																										
[7:0]	RW	test_wrdata				Data write signal of the debugging channel																										



CHIPSET_INFO

CHIPSET_INFO is a chip information register.

	Offset Address	Register Name	Total Reset Value
Bit	0x01E0	CHIPSET_INFO	0x0008_0000
Name	peri_chipset_info		dts_flag reserved dolby_flag
Reset	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 0 0 0 0 0		
Bits	Access	Name	Description
[31:3]	RO	peri_chipset_info	CA chipset_info[31:0]
[2]	RO	dts_flag	DTS support 0: not supported 1: supported
[1]	RO	reserved	Reserved
[0]	RO	dolby_flag	Dolby support 0: supported 1: not supported

PERI_SW_SET

PEIR_SW_SET is a software flag configuration register.

	Offset Address	Register Name	Total Reset Value
Bit	0x01F0	PERI_SW_SET	0x0000_0000
Name	peri_sw_set		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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	peri_sw_set	This register can be written only once.

PERI_SIM_OD_CTRL

PERI_SIM_OD_CTRL is an SCI pin output OD mode control register.



PERI_FEPHY_LDO_CTRL

PERI_FEPHY_LDO_CTRL is an FE PHY LDO control register.

PERI_CPU_A7_CFG

PERI_CPU_A7_CFG is a CPU control register.



	Offset Address	Register Name	Total Reset Value
Bit	0x880	PERI_CPU_A7_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		l2rstdisable l1rstdisable3 l1rstdisable2 l1rstdisable1 l1rstdisable0 reserved func_mbist_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	l2rstdisable	Whether to invalidate the L2 cache automatically when reset on the A7 is deasserted 0x0: yes 0x1: no
[7]	RW	l1rstdisable3	Whether to invalidate the L1 cache automatically when reset on the A7 core 3 is deasserted 0x0: yes 0x1: no
[6]	RW	l1rstdisable2	Whether to invalidate the L1 cache automatically when reset on the A7 core 2 is deasserted 0x0: yes 0x1: no
[5]	RW	l1rstdisable1	Whether to invalidate the L1 cache automatically when reset on the A7 core 1 is deasserted 0x0: yes 0x1: no
[4]	RW	l1rstdisable0	Whether to invalidate the L1 cache automatically when reset on the A7 core 0 is deasserted 0x0: yes 0x1: no
[3:1]	RO	reserved	Reserved
[0]	RW	func_mbist_en	A7 function mbist enable 0x0: disabled 0x1: enabled



PERI_CPU_A7_STAT

PERI_CPU_A7_STAT is a CPU status register.

	Offset Address	Register Name	Total Reset Value
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	PERI_CPU_A7_STAT	0x0000_0000
Name	[31] l2_fun_mbist_tagram_fail [30] l2_fun_mbist_dataram_fail	core3_fun_mbist_ram_fail core2_fun_mbist_ram_fail core1_fun_mbist_ram_fail core0_fun_mbist_ram_fail	fun_mbist_fail fun_mbist_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]	RO	l2_fun_mbist_tagram_fail	L2 tagram function mbist error indicator 0x0: No error occurs. 0x1: Errors occur.
[30]	RO	l2_fun_mbist_dataram_fail	L2 dataram function mbist error indicator 0x0: No error occurs. 0x1: Errors occur.
[29:23]	RO	core3_fun_mbist_ram_fail	RAM error indicator for core 3. The value 1 indicates that errors occur. Bit 29: TLB ram error indicator Bit 28: SCU tagram error indicator Bit 27: L1 ddirty ram error indicator Bit 26: L1 D-cache tagram error indicator Bit 25: L1 D-cache dataram error indicator Bit 24: L1 I-cache tagram error indicator Bit 23: L1 I-cache dataram error indicator
[22:16]	RO	core2_fun_mbist_ram_fail	RAM error indicator for core 2. The value 1 indicates that errors occur. Bit 22: TLB ram error indicator Bit 21: SCU tagram error indicator Bit 20: L1 ddirty ram error indicator Bit 19: L1 D-cache tagram error indicator Bit 18: L1 D-cache dataram error indicator Bit 17: L1 I-cache tagram error indicator Bit 16: L1 I-cache dataram error indicator



[15:9]	RO	core1_fun_mbist_ram_fail	RAM error indicator for core 1. The value 1 indicates that errors occur. Bit 15: TLB ram error indicator Bit 14: SCU tagram error indicator Bit 13: L1 ddirty ram error indicator Bit 12: L1 D-cache tagram error indicator Bit 11: L1 D-cache dataram error indicator Bit 10: L1 I-cache tagram error indicator Bit 9: L1 I-cache dataram error indicator
[8:2]	RO	core0_fun_mbist_ram_fail	RAM error indicator for core 0. The value 1 indicates that errors occur. Bit 8: TLB ram error indicator Bit 7: SCU tagram error indicator Bit 6: L1 ddirty ram error indicator Bit 5: L1 D-cache tagram error indicator Bit 4: L1 D-cache dataram error indicator Bit 3: L1 I-cache tagram error indicator Bit 2: L1 I-cache dataram error indicator
[1]	RO	fun_mbist_fail	A7 function mbist failure indicator 0x0: not failed 0x1: failed
[0]	RO	fun_mbist_done	A7 function mbist completion indicator 0x0: not complete 0x1: complete

PERI_FE_OD_CTRL

PERI_FE_OD_CTRL is an FE LED pin output OD mode control register.

	Offset Address																Register Name								Total Reset Value							
	0x0890																PERI_FE_OD_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																												fe_led_base_od_sel	fe_led_act_od_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		
Bits	Access	Name		Description																												



[31:2]	RW	reserved	Reserved
[1]	RW	fe_led_base_od_sel	Whether FE_LED_BASE is OD output 0: yes 1: no
[0]	RW	fe_led_act_od_sel	Whether FE_LED_ACT is OD output 0: yes 1: no

3.6 Interrupt System

Hi3796M V100 has the A7 and MCU interrupt systems.

Hi3796M V100 uses the GIC integrated in Cortex A7, which supports a maximum of 160 interrupt sources. [Table 3-46](#) describes the interrupt sources of Cortex A7.

Table 3-46 Interrupt sources of Cortex A7

No.	Interrupt Source
0	COMMTX0
1	COMMTX1
2	COMMTX2
3	COMMTX3
4	COMMRX0
5	COMM RX1
6	COMM RX2
7	COMM RX3
8	PMU0
9	PMU1
10	PMU2
11	PMU3
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved



No.	Interrupt Source
17	MCE
18	Reserved
19	Reserved
20	Reserved
21	SWI0_PERI (A9)
22	SWI1_PERI (A9)
23	SWI2_PERI (A9)
24	Timer 0 and timer 1
25	Timer 2
26	Timer 4
27	Timer 6
28	Timer 8/Timer 9 (SEC)
29	Watchdog 0
30	DMAC
31	DDRC0
32	NANDC
33	Reserved
34	SDIO0
35	SDIO1
36	SCI0
37	Reserved
38	I ² C0
39	I ² C1
40	I ² C2
41	Reserved
42	Reserved
43	Reserved
44	Reserved
45	SSP0
46	Reserved
47	IR



No.	Interrupt Source
48	LEDC
49	UART0
50	UART1
51	UART2
52	Reserved
53	Reserved
54	Reserved
55	Reserved
56	Reserved
57	Reserved
58	Timer 3
59	Timer 5
60	Timer 7
61	Reserved
62	USB2HOST1_EHCI
63	USB2HOST1_OHCI
64	Reserved
65	Reserved
66	USB2HOST0_EHCI
67	USB2HOST0_OHCI
68	USB2OTG
69	USB3
70	Reserved
71	ETH
72	Reserved
73	FE_PHY0
74	Reserved
75	Multiplexer
76	Reserved
77	Reserved
78	AXI_SEC_INT



No.	Interrupt Source
79	Reserved
80	Reserved
81	Reserved
82	PVR
83	Reserved
84	AIAO
85	Reserved
86	Reserved
87	Reserved
88	HDMI_TX
89	Reserved
90	VDP
91	TDE
92	Reserved
93	VPSS0
94	GPU
95	VDH_NORM
96	PGD
97	JPGD0
98	Reserved
99	BPD
100	Reserved
101	VEDU
102	JPGE
103	Reserved
104	SCD_NORM
105	VDH_SAFE
106	SCD_SAFE
107	Reserved
108	GPIO0
109	GPIO1



No.	Interrupt Source
110	GPIO2
111	GPIO3
112	GPIO4
113	GPIO5
114	GPIO6
115	Reserved
116	Reserved
117	Reserved
118	Reserved
119	Reserved
120	Reserved
121	Reserved
122	Reserved
123	Reserved
124	Reserved
125	Reserved
126	SEC_INT_CIPHER
127	INT_BPD_SAFE
128	Reserved
129	Reserved
130	Reserved
131	Reserved
132	Reserved
133	Reserved
134	Reserved
135	Reserved
136	Reserved
137	Reserved
138	Reserved
139	Reserved
140	Reserved



No.	Interrupt Source
141	Reserved
142	Reserved
143	Reserved
144	Reserved
145	Reserved
146	Reserved
147	Reserved
148	Reserved
149	Reserved
150	Reserved
151	Reserved
152	Reserved
153	Reserved
154	Reserved
155	Reserved
156	Reserved
157	Reserved
158	Reserved
159	AXI_ERR_IRQ

The chip integrates the MCU, which supports a maximum of 32 interrupt sources, as described in [Table 3-47](#).

Table 3-47 Interrupt sources of the MCU

No.	Interrupt Source
0	WDG
1	Timer 0 and timer 1
2	UART1
3	IR
4	GPIO5/LEDC
5	Timer 2/Timer 3/Timer 4/Timer 5/Timer 6/Timer 7
6	UART0/UART2



No.	Interrupt Source
7	GPIO0/GPIO1/GPIO2/GPIO3/GPIO4/GPIO6
8	Reserved
9	I ² C0/I ² C1/I ² C2
10	SSP0
11	Reserved
12	SCI0
13	Reserved
14	Reserved
15	SDIO0/SDIO1/NANDC
16	USB0_SUSPEND_RESUME/USB1_SUSPEND_RESUME/USB2_SUSPEND_RESUME/USB3_SUSPEND_RESUME
17	USB2HOST0_EHCI/USB2HOST0_OHCI/USB2OTG
18	USB3DRD0
19	Reserved
20	ETH
21	Reserved
22	Multiplexer
23	PVR
24	PGD/JPGD/BPD
25	VDH_SAFE/VDH_NORMAL/SCD_SAFE/SCD_NORMAL
26	VEDU/JPGE
27	VDP/VPSS0
28	TDE/GPU
29	AIAO/HDMI TX
30	DMAC
31	A7TOMCU soft interrupt



3.7 Timer

3.7.1 Overview

The timer module implements the timing and counting functions. It not only provides system clocks for the operating system, but also is used by applications for timing and counting. The timer module has five submodules: dual-timer0, dual-timer 1, dual-timer 2, dual-timer 3, and dual-timer 4. Each dual-timer submodule consists of two timers with the same functions. Besides the different base addresses, the five dual-timer modules provide different functions in the system applications.

- Dual-timer 0 consists of timer 0 and timer 1, which share a base address and an interrupt signal.
- Dual-timer 1 consists of timer 2 and timer 3, which share a base address and an interrupt signal.
- Dual-timer 2 consists of timer 4 and timer 5, which use the same base address but different interrupt signals.
- Dual-timer 3 consists of timer 6 and timer 7, which use the same base address but different interrupt signals.
- Dual-timer 4 consists of timer 8 and timer 9, which share a base address and an interrupt signal.

3.7.2 Features

Each dual-timer module has the following features:

- Provides two 16-/32-bit down counters (each with a programmable 8-bit prescaler).
- Uses the crystal oscillator clock as the count clock.
- Supports three count modes: free-running mode, periodic mode, and one-shot mode.
- Loads the initial value by using the `TIMERx_LOAD` or `TIMERx_BGLOAD` register.
- Allows the current count value to be read at any time.
- Generates an interrupt when the current count value decreases to 0 if the interrupt function is enabled.

3.7.3 Function Description

Typical Application

The timers are provided for software. Each dual-timer submodule uses the fixed 24 MHz count clock. Timer 0 and timer 1 are not powered off in standby mode. They are dedicated for the MCU and cannot be used by the core. Timer 8 and timer 9 in dual-timer 4 share an interrupt signal. The other three timer submodules have the same features.

Function Implementation

The timer works based on a 16-bit or 32-bit (configurable) down counter. The count value decreases by 1 at the rising edge of each count clock. When the count value decreases to 0, the timer generates an interrupt if the interrupt function is enabled.

The timer supports three count modes:

- Free-running mode



The timer continuously counts. When the count value decreases to 0, the timer automatically returns to the maximum value 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, a count value can be loaded, and the count value decreases immediately from the loaded value. When the count value decreases to 0, the maximum value is restored.

- Periodic mode

The timer continuously counts. When the count value decreases to 0, the timer loads the 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 decreases to 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. The working clock of the prescaler can be divided by 1, 16, or 256 in the timer, which provides various frequencies for the count clock.

An initial value can be loaded to the timer in either of the following ways:

- Write to [TIMERx_LOAD](#) to load an initial count value to the timer. If the timer is working, the timer immediately recounts from the new value if a value is written to [TIMERx_LOAD](#). This method applies to all count modes.
- Write to [TIMERx_BGLOAD](#) to set the count cycle in periodic mode. If a value is written to [TIMERx_BGLOAD](#), the timer continues to count until the count value reaches 0, and then loads the written value from [TIMERx_BGLOAD](#) to count.

3.7.4 Working Process

3.7.4.1 Initializing the Timer

The timer must be initialized during system initialization. To initialize timer X ($X = 0\text{--}9$), perform the following steps:

Step 1 Configure [TIMERx_CONTROL](#) to set the timer count length.

Step 2 Write to [TIMERx_LOAD](#) to load an initial value to the timer.

Step 3 (Optional) If you want to enable the timer 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 4 Configure [TIMERx_CONTROL](#) to set the count mode, prescaling divider, and interrupt mask of the timer, and then enable the timer to count.

----End

3.7.4.2 Handling Interrupts

The timer is used to generate interrupts periodically. Therefore, interrupt processing is a process of activating the processes that wait for the timing interrupt. The process is as follows:

Step 1 Configure [TIMERx_INTCLR](#) to clear timer interrupts.

Step 2 Activate the processes that wait for the interrupt.



Step 3 When all the processes that wait for the interrupt are complete or enter the hibernate state, resume the interrupted program.

----End

3.7.5 Register Summary

Table 3-48 describes timer registers.

- Timer 0 and timer 1 share the base address 0xF800_2000.
- Timer 2 and timer 3 share the base address 0xF8A2_9000.
- Timer 4 and timer 5 share the base address 0xF8A2_A000.
- Timer 6 and timer 7 share the base address 0xF8A2_B000.
- Timer 8 and timer 9 share the base address 0xF8A8_1000.

NOTE

The value of x in TIMER x ranges from 0 to 9.

Table 3-48 Summary of timer registers

Offset Address	Register	Description	Page
0x000	TIMER x _LOAD	Initial count value register	3-152
0x004	TIMER x _VALUE	Current count value register	3-152
0x008	TIMER x _CONTROL	Timer control register	3-153
0x00C	TIMER x _INTCLR	Interrupt clear register	3-156
0x010	TIMER x _RIS	Raw interrupt register	3-156
0x014	TIMER x _MIS	Masked interrupt register	3-157
0x018	TIMER x _BGLOAD	Initial count value register in periodic mode	3-158

3.7.6 Register Description

TIMER x _LOAD

TIMER x _LOAD is an initial count value register. It is used to set the initial count value of a timer. Each timer (timer 0–timer 9) has one such register.

When a timer is in periodic mode and the count value decreases to 0, the value of TIMER x _LOAD is reloaded to the counter. When a value is directly written to TIMER x _LOAD, the value of the current counter is changed to the written value on the next rising edge of the TIMCLK enabled by TIMCLKEN x .

NOTE

- The minimum valid value written to TIMER x _LOAD is 1.
- When 0 is written to TIMER x _LOAD, the dual-timer submodule immediately generates an interrupt.



When a value is written to **TIMERx_BGLOAD**, the value of **TIMERx_LOAD** is overwritten, but the current value of the timer count is not affected.

If values are written to **TIMERx_BGLOAD** and **TIMERx_LOAD** before the rising edge of TIMCLK enabled by TIMCLKENx is reached, the value of the counter is changed to the written value of **TIMERx_LOAD** at the next rising edge. From then on, when the counter reaches 0, the last value written to **TIMERx_BGLOAD** or **TIMERx_LOAD** is reloaded.

After **TIMERx_BGLOAD** and **TIMERx_LOAD** are written twice respectively, reading **TIMERx_LOAD** returns the written value of **TIMERx_BGLOAD**.

1. TIMER0_LOAD

Offset Address			Register Name																Total Reset Value									
Bit	0x000			TIMER0_LOAD																0x0000_0000								
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
Bits	Access	Name			Description																							
[31:0]	RW	timer0_load			Initial count value of timer 0																							

2. TIMER1_LOAD

Offset Address			Register Name																Total Reset Value									
Bit	0x020			TIMER1_LOAD																0x0000_0000								
Name				timer1_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
Bits	Access	Name			Description																							
[31:0]	RW	timer1_load			Initial count value of timer 1																							

TIMERx_VALUE

TIMERx_VALUE is a current count value register. It provides the current value of the counter that is counting down.

Each timer (timer 0–timer 9) has one such register.

After a value is written to **TIMERx_LOAD**, **TIMERx_VALUE** immediately reflects the newly loaded value of the counter in the PCLK domain without waiting for the next clock edge of the TIMCLK enabled by TIMCLKENx.

NOTE

When a timer is in 16-bit mode, the upper 16 bits of the 32-bit **TIMERx_VALUE** are not automatically set to 0. If **TIMERx_LOAD** is never written after the timer is switched from 32-bit mode to 16-bit mode, the upper 16 bits of **TIMERx_VALUE** may be non-zero values.



1. TIMER0_VALUE

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	
Bits	Access	Name		Description																												
[31:0]	RO	timer0_value		Current count value of timer 0 that is counting down																												

2. TIMER1_VALUE

Offset Address																Register Name								Total Reset Value								
0x024																TIMER1_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	
Bits	Access	Name		Description																												
[31:0]	RO	timer0_value		Current count value of timer 1 that is counting down																												

TIMERx_CONTROL

TIMERx_CONTROL is a timer control register. Each timer (timer 0–timer 9) 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.



1. TIMER0_CONTROL

Offset Address												Register Name												Total Reset Value											
0x008												TIMER0_CONTROL												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																								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	1	0	0	0	0	0				
Bits	Access	Name		Description																															
[31:8]	-	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		TIMERx_RIS interrupt mask 0: masked 1: not masked																															
[4]	-	reserved		Reserved																															
[3:2]	RW	timerpre		Timer prescaling divider 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: reserved																															
[1]	RW	timersize		Counter select 0: 16-bit counter 1: 32-bit counter																															
[0]	RW	oneshot		Count mode 0: periodic mode or free-running mode 1: one-shot mode																															



2. TIMER1_CONTROL



TIMERx_INTCLR

TIMERx_INTCLR is an interrupt clear register. Writing any value to this register clears the interrupt status of the counter. Each timer (timer 0–timer 9) has one such register.



CAUTION

This register is write-only. When a value is written to this register, the timer interrupts are cleared. No written value is recorded in this register and no default reset value is defined.

1. TIMER0_INTCLR

	Offset Address								Register Name								Total Reset Value															
	0x00C								TIMER0_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	timer0_intclr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	timer0_intclr		Writing to this register clears the output interrupts of timer 0.																												

2. TIMER1_INTCLR

	Offset Address								Register Name								Total Reset Value															
	0x02C								TIMER1_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	timer1_intclr																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	timer1_intclr		Writing to this register clears the output interrupts of timer 1.																												

TIMERx_RIS

TIMERx_RIS is a raw interrupt status register. Each timer (timer 0–timer 9) has one such register.



1. TIMER0_RIS

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		
Bits	Access	Name		Description																												
[31:1]	-	reserved		Reserved Writing to this field has no effect and reading this field returns 0.																												
[0]	RO	timer0ris		Raw interrupt status of timer 0 0: No interrupt is generated. 1: An interrupt is generated.																												

2. TIMER1_RIS

Offset Address																Register Name								Total Reset Value								
0x030																TIMER1_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																												timer1ris			
Reset	0	0	0	0	0	0	0	0	0	0	0	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 Writing to this field has no effect and reading this field returns 0.																												
[0]	RO	timer1ris		Raw interrupt status of timer 1 0: No interrupt is generated. 1: An interrupt is generated.																												

TIMERx_MIS

TIMERx_MIS is a masked interrupt status register. Each timer (timer 0–timer 9) has one such register.



1. TIMER0_MIS

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		
Bits	Access	Name		Description																												
[31:1]	-	reserved		Reserved																												
[0]	RO	timer0mis		Masked interrupt status of timer 0 0: The interrupt is invalid. 1: The interrupt is valid.																												

2. TIMER1_MIS

Offset Address																Register Name								Total Reset Value								
0x034																TIMER1_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																													timer1mis		
Reset	0	0	0	0	0	0	0	0	0	0	0	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	timer1mis		Masked interrupt status of timer 1 0: The interrupt is invalid. 1: The interrupt is valid.																												

TIMERx_BGLOAD

TIMERx_BGLOAD is an initial count value register in periodic mode. Each timer (timer 0–timer 9) has one such register.

This register contains the initial count value of the timer. In periodic mode, this register is used to reload the initial count value when the count value of the timer reaches 0.



This register provides another way for accessing TIMERx_LOAD. After a value is written to TIMERx_BGLOAD, the timer, however, does not count starting from the new written value immediately.

1. TIMER0_BGLOAD

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	0	0	0	0	0		
Bits	Access	Name				Description																																	
[31:0]	RW	timer0bgload				Initial count value of timer 0 Note: This register differs from TIMER0_LOAD. For details, see the description of TIMERX_LOAD.																																	

2. TIMER1_BGLOAD

Offset Address																Register Name																Total Reset Value							
0x038																TIMER1_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	timer1bgload																																						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	timer1bgload				Initial count value of timer 1 Note: This register differs from TIMER1_LOAD. For details, see the description of TIMERX_LOAD.																																	

3.8 DMA Controller

3.8.1 Overview

DMA refers to an operating mode in which I/O data is exchanged by the hardware. In this mode, the DMAC directly transfers data between a memory and a peripheral, between peripherals, or between memories. This avoids the processor intervention and reduces the interrupt handling overhead of the processor. The DMA mode is usually used to transmit data blocks at high speed. After receiving a DMA request, the DMAC enables the bus master controller based on the channel configured by the CPU and sends the address and control signals to the memory and peripherals. The DMAC counts the transmitted data and reports the status of data transfer (data transfer completion or error) to the CPU by using interrupts.



3.8.2 Features

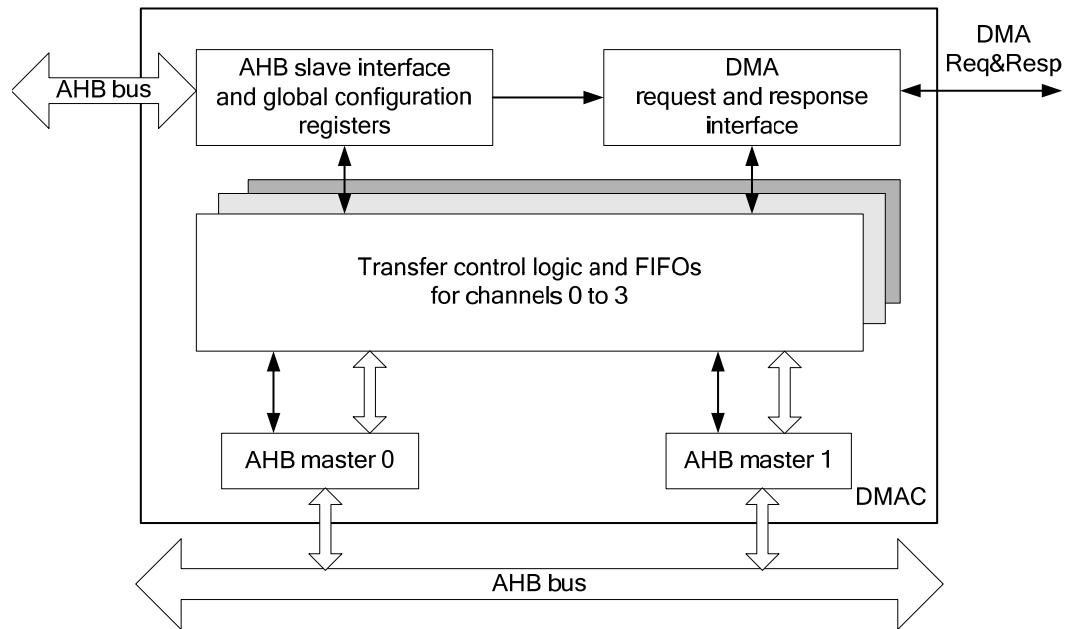
The DMAC has the following features:

- Transfers data in 8-bit, 16-bit, or 32-bit mode.
- Provides four DMA channels. Each channel can be configured for a certain unidirectional transfer.
- Provides fixed DMA channel priorities. DMA channel 0 has the highest priority; whereas channel 3 has the lowest priority. When valid DMA requests from two peripherals are received simultaneously, the channel with the higher priority starts data transfer first.
- Provides a 4 x 32-bit FIFO in each channel of DMAC channels 0 to 3.
- Provides two 32-bit master bus interfaces for data transfer.
- Supports two types of DMA requests for peripherals: single transfer request and burst transfer request.
- Provides 16 groups of DMA request inputs. These requests can be configured as source requests or destination requests.
- Supports the DMA requests controlled by software.
- Programs the DMA burst size.
- Configures the source address and the destination address as automatic incremental or decremental addresses 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.
- Provides a maskable interrupt output and supports the query of the states of the raw/masked DMA error interrupt and DMA transfer completion interrupt and the state of the combination of the two interrupts.
- Controls the power consumption by disabling the DMAC and supports DMAC clock gating.

3.8.3 Function Description

Function Block Diagram

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

Figure 3-3 Functional block diagram of the DMAC

Each DMAC channel involves a group of transfer control logic and one FIFO. The transfer control logic performs the following operations automatically:

- Step 1** Read data from the source address specified by the software.
- Step 2** Buffer the data to the FIFO.
- Step 3** Fetch data from the FIFO.
- Step 4** Write the data to the destination address specified by the software.

----End

Working Process

The workflow of the DMAC is as follows:

- Step 1** The software selects one DMAC channel for DMA transfer, configures the following items, and enables the channel:
 - Source address
 - Destination address
 - Head pointer of the linked list
 - Amount of the transferred data
 - Source and destination peripheral request signal numbers
 - Masters used at the source and destination ends of the channel.

After the channel is enabled, the DMAC starts to check the activities of the DMA request signals between the source peripheral and destination device connected to this channel.

- Step 2** The source device transmits 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.

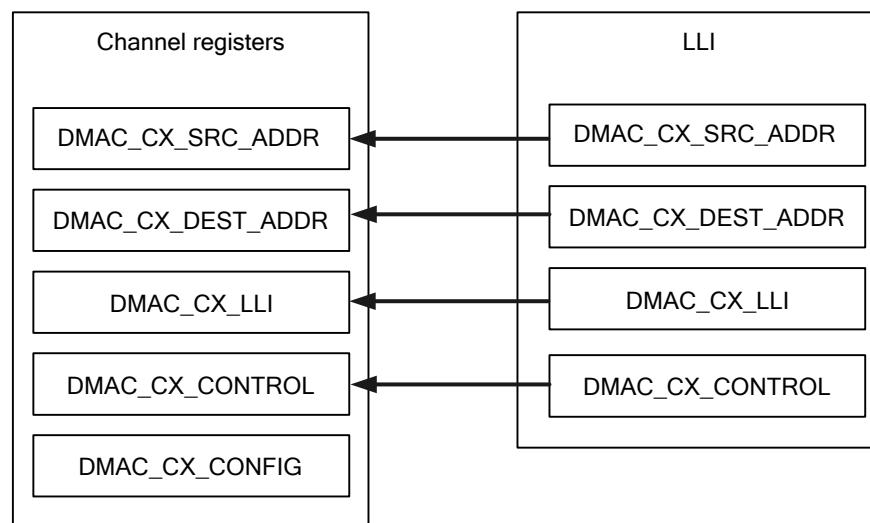
- Step 3** The DMAC channel responds to the DMA request of the source device, reads data from the source device, and stores the data to the internal FIFO of the channel.
- Step 4** The destination device sends a DMA request to the DMAC. If the destination device is a memory, the DMAC considers that the DMA request is always valid by default.
- Step 5** The DMAC channel responds to the DMA request of the destination device, fetches data from the internal FIFO of the channel, and writes it to the destination device.

Steps 2 and 3 and steps 4 and 5 may be performed concurrently, because the source and destination devices may initiate DMA requests to the DMAC at the same time. When the FIFO overrun or underrun occurs on the DMA channel, the DMAC blocks the DMA requests of the source device or destination device until the FIFO is not full or empty. When the DMAC interacts with the source device and destination device, step 2 to step 5 are performed repeatedly until the specified data is transferred and a transfer completion interrupt (maskable) is generated. If the value of [DMAC_CXLLI](#) is not 0, read linked list item (LLI) nodes by considering the value of [DMAC_CXLLI](#) as an address, load the read values to [DMAC_CX_SRC_ADDR](#), [DMAC_CX_DEST_ADDR](#), [DMAC_CXLLI](#), and [DMAC_CX_CONTROL](#) in sequence (see [Figure 3-4](#)), and then go to step 2. If the value of [DMAC_CXLLI](#) is 0, the current DMA transfer is stopped. In this case, the channel is disabled automatically and the data transfer ends.

----End

[Figure 3-4](#) shows how to update channel registers by using LLI.

Figure 3-4 Updating channel registers by using LLI



Connection Between the DMAC and Peripherals

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

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

- DMACBREQ

Burst transfer request signal. It triggers a burst transfer and the burst size is preconfigured.

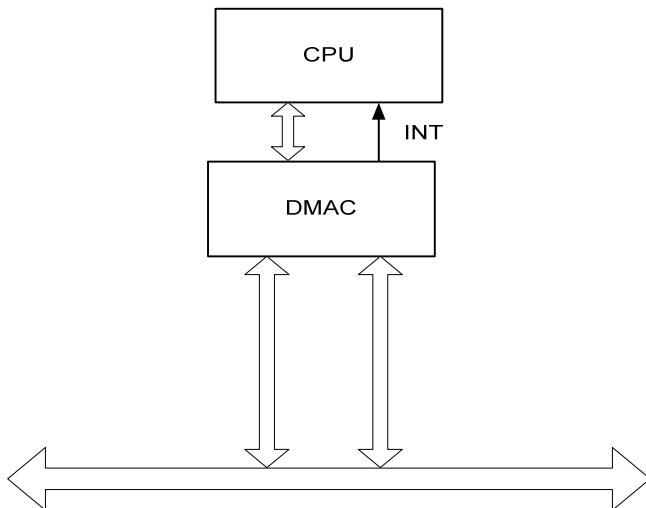
- DMACSREQ

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.

The DMAC provides a request clear signal DMACLR. This signal is sent to each peripheral by the DMAC as response to the DMA request signal of each peripheral.

[Figure 3-5](#) shows the connections between the DMAC and other modules.

Figure 3-5 Connections between the DMAC and other modules



3.8.4 Operating Mode

Clock Gating

The DMAC and DMAC clock can be disabled using software in the following scenarios to reduce power consumption:

- All DMAC channels are idle and there is no DMA transfer request.
- [DMAC_CX_CONFIG\[e\]](#) is set to 0 and the DMAC channel is disabled.

To disable the DMAC clock, perform the following steps:

Step 1 Write 0 to [DMAC_CX_CONFIG\[e\]](#) to disable DMAC channels.

Step 2 Write 0 to [DMAC_CONFIG\[e\]](#) to disable the DMAC.

Step 3 Write 0 to [PERI_CRG33\[dma_cken\]](#) to disable the DMAC clock.

Enable the clock and the DMAC again when the DMAC is required for data transfer.

----End

Initializing the IDMAC

To initialize the DMAC, perform the following steps:



- Step 1** Write to **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_ERR_CLR** and **DMAC_INT_TC_CLR** to clear all interrupts.
- 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 by writing 0 to **DMAC_CX_CONFIG[e]** of each channel.

----End

Enabling a Channel

After the DMAC is initialized, it can transmit data only when a DMAC channel is configured and enabled. To enable a DMA channel, perform the following steps:

- Step 1** Read **DMAC_ENBLD_CHNS** to search for idle channels and select one.
- Step 2** Write 1 to the corresponding bits of **DMAC_INT_ERR_CLR** and **DMAC_INT_TC_CLR** to clear the interrupt status of the selected channel.
- Step 3** Configure and enable the selected channel as follows:
1. Write to **DMAC_CX_SRC_ADDR** to set the access start address of the source device.
 2. Write to **DMAC_CX_DEST_ADDR** to set the access start address of the destination device.
 3. For single-block data transfer, write 0 to **DMAC_CXLLI**.
 4. For LLI data transfer, set **DMAC_CXLLI** to 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 destination device.
 6. Write to **DMAC_CX_CONFIG** to set the DMA request signal, flow control mode, and interrupt mask of the channel. At this time, write 0 to **DMAC_CX_CONFIG[e]**, that is, this channel is not enabled currently.
 7. Write to **DMAC_CX_CONFIG** to enable this channel. Set the Channel Enable bit to 1 and keep other bits unchanged.

----End

Handling Interrupts

When data transfer is complete or an error occurs during data transfer, an interrupt is reported to the interrupt controller. The interrupt is handled as follows:

- Step 1** Read **DMAC_INT_STAT** to find the channel that transmits an interrupt request. When multiple channels initiate interrupt requests at the same time, the interrupt request with the highest priority is handled first.
- Step 2** Read **DMAC_INT_TC_STAT** to query the value of the selected bit to check whether the interrupt sent by the corresponding channel is a transfer completion interrupt. If the value is 1, the interrupt is a transfer completion interrupt. In this case, go to step 4; otherwise, go to step 3.



Step 3 Read **DMAC_INT_ERR_STAT** to query the value of the selected bit to check whether the interrupt sent by the corresponding channel is an error interrupt. If the selected bit is 1, the interrupt is an error interrupt. In this case, go to step 5; otherwise, end the operation.

Step 4 Handle the transfer completion interrupt as follows:

1. Write 1 to the selected bit of **DMAC_INT_TC_CLR** to clear the interrupt status of the corresponding channel.
2. Fetch or use up the data buffered in the memory. If necessary (for example, a buffer needs to be created in the memory), configure and enable the channel again.
3. End interrupt operations.

Handle the error interrupt as follows:

1. Write 1 to the selected bit of **DMAC_INT_ERR_CLR** to clear the interrupt status of the corresponding channel.
2. Provide the error information. If necessary, configure and enable the channel again.
3. End interrupt operations.

----End

3.8.5 Register Summary

Table 3-49 describes the DMAC registers.



The variable *X* in the offset addresses of registers indicates the channel ID and ranges from 0 to 3.

Table 3-49 Summary of DMAC registers (base address: 0xF987_0000)

Offset Address	Register	Description	Page
0x000	DMAC_INT_STAT	Interrupt status register	3-166
0x004	DMAC_INT_TC_STAT	DMAC transfer completion interrupt status register	3-167
0x008	DMAC_INT_TC_CLR	Transfer completion status clear register	3-167
0x00C	DMAC_INT_ERR_STAT	Error interrupt status register	3-168
0x010	DMAC_INT_ERR_CLR	Error interrupt clear register	3-168
0x014	DMAC_RAW_INT_TC_STATUS	Raw transfer completion interrupt status register	3-168
0x018	DMAC_RAW_INT_ERR_STATUS	Raw transfer error interrupt status register	3-169
0x01C	DMAC_ENBLD_CHNS	Channel enable register	3-169
0x020	DMAC_SOFT_BREQ	Software burst request register	3-170
0x024	DMAC_SOFT_SREQ	Software single request register	3-171



Offset Address	Register	Description	Page
0x028	DMAC_SOFT_LBREQ	Software last burst request register	3-171
0x02C	DMAC_SOFT_LSREQ	Software last single request register	3-172
0x030	DMAC_CONFIG	Configuration register	3-172
0x034	DMAC_SYNC	Sync register	3-173
0x100 + X x 0x20	DMAC_CX_SRC_ADDR	DMAC source address register	3-174
0x104 + X x 0x20	DMAC_CX_DEST_ADDR	DMAC destination address register	3-174
0x108 + X x 0x20	DMAC_CXLLI	DMAC LLI register	3-175
0x10C + X x 0x20	DMAC_CX_CONTROL	Channel control register	3-176
0x110 + X x 0x20	DMAC_CX_CONFIG	Channel configuration register	3-181

3.8.6 Register Description

DMAC_INT_STAT

DMAC_INT_STAT is an interrupt status register. It shows the masked DMAC interrupt status. If 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 also masked. Each bit of DMAC_INT_STAT maps to one DMAC channel. When a bit is 1, an interrupt request is generated in the corresponding channel and the interrupt request may be an error interrupt or a transfer completion interrupt.

	Offset Address	Register Name	Total Reset Value
Bit	0x000	DMAC_INT_STAT	0x0000_0000
Name	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	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
[31:4]	-	reserved	Reserved
[3:0]	RO	int_stat	Masked interrupt status of each DMA channel. Bit[3:0] map to channels 3–0. 0: No interrupt is generated. 1: An interrupt is generated. The interrupt request may be an error interrupt or a transfer completion interrupt.



DMAC_INT_TC_STAT

DMAC_INT_TC_STAT is a transfer completion interrupt status register. It shows the status of the masked transfer completion interrupts. The corresponding mask bit is DMAC_CX_CONFIG[itc], where X is the channel number ranging from 0 to 3. This register must work with the DMAC_INT_STAT register.

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		
Bits	Access	Name			Description																											
[31:4]	-	reserved			Reserved																											
[3:0]	RO	int_tc_stat			Masked transfer completion interrupt status. Bit[3:0] map to channels 3–0. 0: No transfer completion interrupt is generated. 1: A transfer completion interrupt is generated.																											

DMAC_INT_TC_CLR

DMAC_INT_TC_CLR is a transfer completion interrupt clear register. It is used to clear transfer completion interrupts.

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		
Bits	Access	Name			Description																											
[31:4]	-	reserved			Reserved																											
[3:0]	WO	int_tc_clr			Transfer completion interrupt clear. Bit[3:0] map to channels 3–0. 0: not cleared 1: cleared																											



DMAC_INT_ERR_STAT

DMAC_INT_ERR_STAT is an error interrupt status register. It shows the masked error interrupt status. The corresponding mask bit is DMAC_CX_CONFIG[ie].

Offset Address																Register Name								Total Reset Value							
Bit	0x00C															DMAC_INT_ERR_STAT								0x0000_0000							
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	
Bits	Access	Name			Description																										
[31:4]	-	reserved			Reserved																										
[3:0]	RO	int_err_stat			Masked error interrupt status. Bit[3:0] map to channels 3–0. 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.

Offset Address																Register Name								Total Reset Value							
Bit	0x010															DMAC_INT_ERR_CLR								0x0000_0000							
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	
Bits	Access	Name			Description																										
[31:4]	-	reserved			Reserved																										
[3:0]	WO	int_err_clr			Error interrupt clear. Bit[3:0] map to channels 3–0. 0: not cleared 1: cleared																										

DMAC_RAW_INT_TC_STATUS

DMAC_RAW_INT_TC_STATUS is a raw transfer completion interrupt status register. It shows the status of the raw transfer completion interrupt of each channel.



Offset Address																Register Name								Total Reset Value								
0x014																DMAC_RAW_INT_TC_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																									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	
Bits	Access	Name			Description																											
[31:4]	-	reserved			Reserved																											
[3:0]	RO	raw_int_tc_stat			Raw transfer completion interrupt status. Bit[3:0] map to channels 3–0. 0: No transfer completion interrupt is generated. 1: A transfer completion interrupt is generated.																											

DMAC_RAW_INT_ERR_STATUS

DMAC_RAW_INT_ERR_STATUS is a 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_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																									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		
Bits	Access	Name			Description																											
[31:4]	-	reserved			Reserved																											
[3:0]	RO	raw_int_err_stat			Raw error interrupt status of each channel. Bit[3:0] map to channels 3–0. 0: No error interrupt is generated. 1: An error interrupt is generated.																											

DMAC_ENBLD_CHNS

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

If a bit of DMAC_ENBLD_CHNS is 1, the corresponding channel is enabled. An enable bit in the DMAC_CX_CONFIG register determines whether a corresponding channel is enabled. If the DMA transfer over a channel is complete, the corresponding bit in the DMAC_ENBLD_CHNS register 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		
Bits	Access	Name			Description																											
[31:4]	-	reserved			Reserved																											
[3:0]	RO	enabled_channels			Channel enable. Bit[3:0] map to channels 3–0. 0: disabled 1: enabled																											

DMAC SOFT BREQ

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

Reading this register queries the device that is requesting the DMA burst transfer. This register and any peripheral each can generate a DMA request.



NOTE

You are advised not to 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		
Bits	Access	Name			Description																											
[31:16]	-	reserved			Reserved																											
[15:0]	RW	soft_breq			<p>Whether to generate DMA burst requests by the software</p> <p>When this register is written:</p> <p>0: no effect</p> <p>1: A DMA burst transfer request is generated. When the transfer is complete, the corresponding bit is cleared.</p> <p>When this register is read:</p> <p>0: The peripheral corresponding to the request signal DMACBREQ[15:0] does not send a DMA burst request.</p> <p>1: The peripheral corresponding to the request signal DMACBREQ[15:0] is requesting a DMA burst transfer.</p>																											

DMAC_SOFT_SREQ

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

Reading this register queries the device that is requesting the DMA single transfer. This register and any of the 16 DMA request input signals of the DMAC each can generate a DMA request.

 NOTE

You are advised not to use a software DMA request and a hardware DMA request at the same time.

DMAC_SOFT_LBREQ

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



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		
Bits	Access	Name				Description																										
[31:16]	-	reserved				Reserved																										
[15:0]	WO	soft_lbreq				Whether to generate a DMA last burst transfer request by the software 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 a software last single request register. It is used for the software to control whether to generate a DMA last single transfer request.

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		
Bits	Access	Name				Description																										
[31:16]	-	reserved				Reserved																										
[15:0]	WO	soft_lsreq				Whether to generate a DMA last single transfer request by the software 0: no effect 1: A DMA last single transfer request is generated. When the transfer is complete, the corresponding bit is cleared.																										

DMAC_CONFIG

DMAC_CONFIG is a configuration register for configuring the DMAC. You can change the endianness mode of the two master interfaces of the DMAC by writing to 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	
Bits	Access	Name		Description																												
[31:3]	-	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 a synchronization register. It is used to enable or disable the synchronization logic provided for DMA 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	
Bits	Access	Name		Description																												
[31:16]	-	reserved		Reserved																												
[15:0]	RW	dmac_sync		Sync logic enable for DMA request signals of the corresponding peripheral 0: enabled 1: disabled																												



DMAC_CX_SRC_ADDR

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

Its offset address is $0x100 + X \times 0x20$. X ranges from 0 to 3, corresponding to DMA channels 0 to 3.

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

- The source address is incremented.
- A complete data block is transferred and then loaded from LLI nodes.

If a channel is active, no valid information can be obtained when this register is read. This is because that after software obtains the register value, the value has changed during data transfer. Therefore, this 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 destination addresses must be aligned with the transfer widths of the source and destination devices.

Offset Address																Register Name								Total Reset Value									
0x100 + X x 0x20																DMAC_CX_SRC_ADDR								0x0000_0000									
(X = 0-3)																																	
Bit	31	30	29	28	27	26	25	24	23	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																												

DMAC_CX_DEST_ADDR

DMAC_CX_DEST_ADDR is a destination address register. Its offset address is $0x104+X*0x20$. X ranges from 0 to 3, corresponding to DMA channels 0 to 3.

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

- The destination address is incremented.
- A complete data block is transferred and then loaded from LLI nodes.

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



Offset Address 0x104 + X * 0x20 (X = 0-3)																Register Name DMAC_CX_DEST_ADDR								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	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			
Bits	Access	Name		Description																												
[31:0]	RW	dest_addr		DMA destination address																												

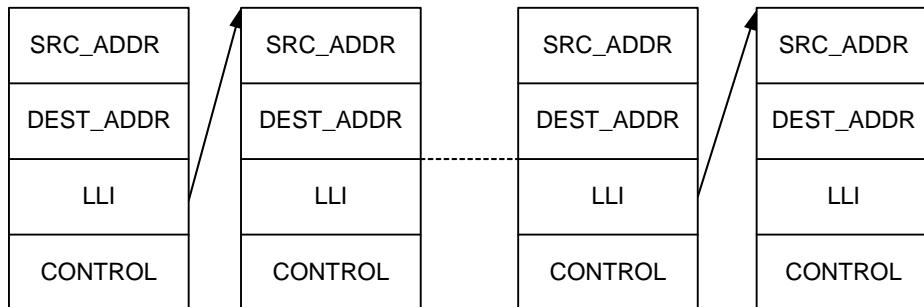
DMAC_CXLLI

DMAC_CXLLI is an LLI register. Its offset address is $0x108+X*0x20$. X ranges from 0 to 3, corresponding to DMA channels 0 to 3.

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

- [DMAC_CX_SRC_ADDR](#) specifies the start address of the source device
- [DMAC_CX_DEST_ADDR](#) specifies the start address of the destination device.
- [DMAC_CX_LLNI](#) specifies the address of the next node
- [DMAC_CX_CONTROL](#) specifies the master, data width, burst size, address increment, and transfer size of the source device and destination device.

Figure 3-6 Structure of the linked list for the DMAC



CAUTION

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

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



Offset Address 0x108+X*0x20 (X = 0-3)																Register Name DMAC_CXLLI								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	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			
Bits	Access	Name		Description																												
[31:2]	RW	lli		LLI. Bit[31:2] in the next LLI node address and the address bit[1:0] are set to 0. A linked list address must be 4-byte aligned.																												
[1]	RW	reserved		Reserved. This bit value must be set to 0 during write operations and masked during read operations.																												
[0]	RW	lm		Master for loading the next LLI node 0: master 1 1: master 2																												

DMAC_CX_CONTROL

DMAC_CX_CONTROL is a channel control register. Its offset address is 0x10C+I*0x20. X ranges from 0 to 3, corresponding to DMA channels 0 to 3.

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 corresponding register must be programmed by software. After the channel is enabled, the register is updated when being loaded from a linked list node after a complete data block is transferred.

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



Offset Address 0x10C+X*0x20 (X = 0~3)												Register Name DMAC_CX_CONTROL								Total Reset Value 0x0000_0000														
Bit	31	30	29	28	27	26	25	24	23	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			
Bits	Access	Name		Description																														
[31]	RW	i		Transfer completion interrupt enable This bit determines whether the current LLI node triggers a transfer completion interrupt. 0: disabled 1: enabled																														
[30:28]	RW	prot		H PROT[2:0] access protection signal transmitted by the master																														
[27]	RW	di		Destination address increment 0: The destination address is not incremented. 1: The destination address is incremented each time a data segment is transferred. If the destination device is a peripheral, the destination address is not incremented. If the destination device is a memory, the destination address is incremented.																														
[26]	RW	si		Source address increment 0: The source address is not incremented. 1: The source address is incremented each time 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 destination device 0: master 1 for accessing the SPDIF, SIO0, SIO1, UART0, UART1, and SCI 1: master 2 for accessing the SPI flash, NAND flash, NOR flash, and DDRC																														
[24]	RW	s		Master for accessing the source device 0: master 1 for accessing the SPDIF, SIO0, SIO1, UART0, UART1, and SCI 1: master 2 for accessing the SPI flash, NAND flash, and DDRC																														



[23:21]	RW	dwidth	<p>Transfer bit width of the destination device</p> <p>The transfer bit width is invalid if it is greater than the bit width of the master.</p> <p>The data widths of the destination and source devices can be different. The hardware automatically packs and unpacks the data.</p> <p>For details about the mapping between the value of DWidth and the bit width, see Table 3-51.</p>
[20:18]	RW	swidth	<p>Transfer bit width of the source device</p> <p>The transfer bit width is invalid if it is greater than the bit width of the master.</p> <p>The data widths of the destination and source devices can be different. The hardware automatically packs and unpacks the data.</p> <p>For details about the mapping between the value of SWidth and the bit width, see Table 3-51.</p>
[17:15]	RW	dbsize	<p>Burst size of the destination device, indicating the number of data segments to be transferred by the destination device in a burst transfer, that is, the number of transferred data segments when DMACCxBREQ is valid</p> <p>It must be set to a burst size supported by the destination device. If the destination device is a memory, set it to the storage area size beyond the storage address boundary.</p> <p>For details about the mapping between the value of DBSize and the transfer size, see Table 3-50.</p>
[14:12]	RW	sbsize	<p>Burst size of the source device, indicating 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</p> <p>It must be set to a burst size supported by the source device. If the source device is a memory, set it to the storage area size beyond the storage address boundary.</p> <p>For details about the mapping between the value of SBSIZE and the transfer size, see table Table 3-51.</p>
[11:0]	RW	transfersize	<p>The DMA transfer size can be configured by writing to this register only when the DMAC is a flow controller. This field indicates the amount of data to be transferred by the source device.</p> <p>When this register is read, the amount of data transferred through the bus connected to the destination device is obtained.</p> <p>If a channel is active, no valid information can be obtained when this register is read. This is because that after software obtains the register value, the value has changed during data transfer.</p> <p>Therefore, this register is read after the channel is enabled and data transfer stops.</p>

[Table 3-50](#) describes the mapping between DBSize or SBSIZE of [DMAC_CX_CONTROL](#) and the burst lengths.



Table 3-50 Mapping between the value of DBSize or SBSIZE and the burst size

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

Table 3-51 describes the mapping between DWidth or SWidth of **DMAC_CX_CONTROL** and transfer data width.

Table 3-51 Mapping between DWidth or SWidth and the transfer bit width

SWidth or DWidth	Transfer Bit Width
000	Byte (8 bits)
001	Half-word (16 bits)
010	Word (32 bits)
011	Reserved
100	Reserved
101	Reserved
110	Reserved
111	Reserved

Note the following when configuring **DMAC_CX_CONTROL**:

- When the transfer bit width of the source device is smaller than that of the destination device, the product of the transfer bit width and transfer size of the source device must be an integral multiple of the transfer bit width of the destination device. Otherwise, data retention and data loss occur in the FIFO.
- The 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 different from the common register field whose written value and read value may be the same. During writes, this field serves as a control



register, because it determines the number of data segments to be 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 but not the DMAC FIFO) of the source device or destination device, the mode of DMAC source address or destination address must be set to non-increment mode. Otherwise, the peripheral FIFO may overflow.

The bus access information is provided for the source device or destination device over the master interface signals during data transfer. Such information is related to the bits [DMAC_CX_CONTROL\[prot\]](#) and [DMAC_CX_CONTROL\[Lock\]](#) configured by programming channel registers. [Table 3-52](#) describes the three protection bits of the prot field of [DMAC_CX_CONTROL](#).

Table 3-52 Definitions of the prot field of [DMAC_CX_CONTROL](#)

Bits	Description	Purpose
[2]	Cacheable or noncacheable	Specifies whether the access is cacheable. 0: not cacheable 1: cacheable For example, this bit can notify an advanced microcontroller bus architecture (AMBA) bridge of the following information: When finding the first read operation of the 8-digit burst, this bridge can initiate one 8-digit burst read operation on the destination bus directly instead of transmitting the read operations on the source bus to the destination bus one by one. This bit controls the output of the bus signal HPROT[3].
[1]	Bufferable or nonbufferable	Specifies whether the access is bufferable. 0: not bufferable 1: bufferable For example, this bit can notify an AMBA bridge that the write operation on the source bus can be complete without waiting. That is, the operation can be performed even when the bridge does not arbitrate the operation to the destination bus and the slave device does not receive data completely. This bit controls the output of the bus signal HPROT[2].
[0]	Privileged or User	Specifies the 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 a channel configuration register. Its offset address is $0x110 + X \times 0x20$. X ranges from 0 to 3, corresponding to DMA channels 0 to 3.

This register is not updated when a new linked list node is loaded.

Offset Address		Register Name																Total Reset Value																
0x110+X*0x20 (X = 0-3)		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																																	
Bits	Access	Name		Description																														
[31:19]	-	reserved		Reserved This bit must be set to 0 during write operations and masked during read operations.																														
[18]	RW	h		Halt bit 0: The DMA request is allowed. 1: The subsequent DMA requests are ignored and data in the channel FIFO is completely transmitted. This bit can work with the Active bit and the Channel Enable bit to disable a DMA channel without data loss.																														
[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 work with the Halt bit and the Channel Enable bit to disable a DMA channel without data loss.																														
[16]	RW	l		Lock bit 0: Disable lock transfer on the bus. 1: Enable lock transfer on the bus.																														
[15]	RW	itc		Transfer completion interrupt mask 0: masked 1: not masked																														
[14]	RW	ie		Transfer error interrupt mask 0: masked 1: not masked																														



[13:11]	RW	flow_cntrl	<p>Flow control and transfer type</p> <p>This field specifies the flow controller and transfer type. The flow controller can be the DMAC, source device, or destination device. The transfer type can be memory to peripheral, peripheral to memory, peripheral to peripheral, or memory to memory. For details, see Table 3-53.</p>
[10]	-	reserved	<p>Reserved</p> <p>This bit must be set to 0 during write operations and masked during read operations.</p>
[9:6]	RW	dest_peripheral	<p>Destination device</p> <p>The field is used to select a peripheral request signal as the request signal for the DMA destination device of the channel.</p> <p>If the destination device for DMA transfer is a memory, this field is ignored.</p>
[5]	-	reserved	<p>Reserved</p> <p>This bit must be set to 0 during write operations and masked during read operations.</p>
[4:1]	RW	src_peripheral	<p>Source device</p> <p>This field is used to select a peripheral request signal as the request signal for the DMA source device of the channel.</p> <p>If the source device for DMA transfer is a memory, this field is ignored.</p>
[0]	RW	e	<p>Channel enable</p> <p>Reading this field obtains the current channel status.</p> <p>0: disabled</p> <p>1: enabled</p> <p>Clearing this bit can disable a channel. When this bit is cleared, the current bus transfer continues until the data transfer is complete. Then, the channel is disabled and the remaining data in the FIFO is lost. When the last LLI node 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 that the subsequent DMA requests are ignored by the channel. After this, the Active bit must be polled until its value becomes 0, indicating that there is no data in the channel FIFO. At this time, the Enable bit can be cleared.</p> <p>To enable a channel by setting this bit to 1, you must reinitialize the channel. If a channel is enabled by setting this bit to 1 only, an unexpected result may occur.</p>

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. The running delay during a bus burst operation also needs to be considered.



[Table 3-53](#) describes the flow controllers and transfer types corresponding to the flow_cntrl field of [DMAC_CX_CONFIG](#).

Table 3-53 Flow controllers and transfer types corresponding to the flow_cntrl field

Bit Value	Transfer Mode	Controller
000	Memory to memory	DMAC
001	Memory to peripheral	DMAC
010	Peripheral to memory	DMAC
011	Source device to destination device	DMAC
100	Source device to destination device	Destination device
101	Memory to peripheral	Destination device
110	Peripheral to memory	Source device
111	Source device to destination device	Source device



Contents

4 Power Management	4-1
4.1 Overview	4-1
4.2 Operating Modes	4-1
4.3 Clock Gating and Clock Frequency Adjustment	4-2



4 Power Management

4.1 Overview

Hi3796M V100 provides various low-power control modes to dynamically reduce its power consumption.

- System operating mode control
In each operating mode except the normal mode, power consumption is reduced to some extent. You can select an operating mode based on the actual power consumption and function requirements.
- Clock gating and clock frequency adjustment
Unnecessary clocks can be disabled to reduce power consumption. When the function requirements are met, the clock frequency can be adjusted to dynamically reduce power consumption.
- Module low-power control
When a module is idle, it can be disabled or switched to low-power mode to reduce power consumption.
- DVFS and AVS based on CPU load monitoring

4.2 Operating Modes

The system provides two operating modes:

- Normal mode: operating mode when the system runs normal functional services
When the system works in normal mode, power consumption can be reduced by using DVFS or AVS or disabling unused clocks.
- Standby mode (light standby mode and passive standby mode)
 - In light standby mode, clocks of most modules such as the CPU and GPU are disabled, and wakeup over the network or USB port is supported.
 - In passive standby mode, the board power supply is turned off. Scheduled wakeup by using the IR or wakeup by pressing a key on the front panel is supported.



4.3 Clock Gating and Clock Frequency Adjustment

The system supports clock gating of each module. When a module is idle, its clock can be disabled to reduce power consumption.

The working frequencies of some modules can also be separately adjusted to reduce power consumption. For details about the clock source of each module, see section 3.3 "Clock."



Contents

5 Memory Interfaces	5-1
5.1 DDRC.....	5-1
5.1.1 Overview.....	5-1
5.1.2 Features.....	5-1
5.1.3 Function Description.....	5-2
5.2 NANDC	5-11
5.2.1 Overview.....	5-11
5.2.2 Features.....	5-11
5.2.3 Function Description.....	5-12
5.2.4 Data Structure	5-17
5.2.5 Software Operations.....	5-24
5.2.6 Others.....	5-26
5.3 MMC/SD/SDIO Controller.....	5-26
5.3.1 Function Description.....	5-27
5.3.2 Timings and Parameters	5-34
5.3.3 Application.....	5-35



Figures

Figure 5-1 Connecting the DDRC to two 16-bit DDR3 SDRAMs in 32-bit mode	5-3
Figure 5-2 Connecting the DDRC to four 8-bit DDR3 SDRAMs in 32-bit mode	5-4
Figure 5-3 Connecting the DDRC to a 16-bit DDR3 SDRAM in 16-bit mode	5-5
Figure 5-4 Connecting the DDRC to two 8-bit DDR3 SDRAMs in 16-bit mode	5-6
Figure 5-5 Block diagram of NANDC interfaces	5-13
Figure 5-6 Typical timing when the NANDC reads a page of data from the NAND flash	5-13
Figure 5-7 Typical timing when the NANDC programs the NAND flash	5-14
Figure 5-8 Data structure of the driver in 8-bit ECC mode (2 KB page size)	5-17
Figure 5-9 Data structure of data in the NAND flash (with the 2 KB page size) in 8-bit ECC mode	5-17
Figure 5-10 Data structure of the driver in 8-bit ECC mode (4 KB page size)	5-17
Figure 5-11 Data structure of data in the NAND flash (with the 4 KB page size) in 8-bit ECC mode	5-18
Figure 5-12 Data structure of the driver in 16-bit ECC mode (2 KB page size)	5-18
Figure 5-13 Data structure of data in the NAND flash (with the 2 KB page size) in 16-bit ECC mode	5-18
Figure 5-14 Data structure of the driver in 16-bit ECC mode (4 KB page size)	5-18
Figure 5-15 Data structure of data in the NAND flash (with the 4 KB page size) in 16-bit ECC mode	5-19
Figure 5-16 Data structure of the driver in 24-bit ECC mode (2 KB page size)	5-19
Figure 5-17 Data structure of data in the NAND flash (with the 2 KB page size) in 24-bit ECC mode	5-19
Figure 5-18 Data structure of the driver in 24-bit ECC mode (4 KB page size)	5-19
Figure 5-19 Data structure of data in the NAND flash (with the 4 KB page size) in 24-bit ECC mode	5-20
Figure 5-20 Data structure of the driver in 24-bit ECC mode (8 KB page size)	5-20
Figure 5-21 Data structure of data in the NAND flash (with the 8 KB page size) in 24-bit ECC mode	5-20
Figure 5-22 Data structure of the driver in 28-bit ECC mode (2 KB page size)	5-20
Figure 5-23 Data structure of data in the NAND flash (with the 2 KB page size) in 28-bit ECC mode	5-21
Figure 5-24 Data structure of the driver in 28-bit ECC mode (4 KB page size)	5-21
Figure 5-25 Data structure of data in the NAND flash (with the 4 KB page size) in 28-bit ECC mode	5-21
Figure 5-26 Data structure of the driver in 28-bit ECC mode (8 KB page size)	5-21



Figure 5-27 Data structure of data in the NAND flash (with the 8 KB page size) in 28-bit ECC mode	5-22
Figure 5-28 Data structure of the driver in 40-bit ECC mode (8 KB page size)	5-22
Figure 5-29 Data structure of data in the NAND flash (with the 8 KB page size) in 40-bit ECC mode	5-22
Figure 5-30 Data structure of the driver in 40-bit ECC mode (16 KB page size)	5-22
Figure 5-31 Data structure of data in the NAND flash (with the 16 KB page size) in 40-bit ECC mode	5-23
Figure 5-32 Data structure of the driver in 64-bit ECC mode (8 KB page size)	5-23
Figure 5-33 Data structure of data in the NAND flash (with the 8 KB page size) in 64-bit ECC mode	5-23
Figure 5-34 Data structure of the driver in 64-bit ECC mode (16 KB page size)	5-24
Figure 5-35 Data structure of data in the NAND flash (with the 16 KB page size) in 64-bit ECC mode	5-24
Figure 5-36 Functional block diagram of the MMC controller	5-27
Figure 5-37 Typical application circuit of the MMC controller	5-28
Figure 5-38 Format of an MMC command	5-29
Figure 5-39 Format of the response to an MMC command	5-30
Figure 5-40 Running an MMC non-data transfer command	5-30
Figure 5-41 Single-block and multi-block read operations	5-32
Figure 5-42 Single-block and multi-block write operations	5-32
Figure 5-43 Data transfer format in 1-bit mode	5-33
Figure 5-44 Data transfer format in 4-bit mode	5-33
Figure 5-45 Output timing	5-35
Figure 5-46 Input timing	5-35
Figure 5-47 Structure of the dual-buffer descriptor	5-42
Figure 5-48 Structure of the linked descriptor	5-43
Figure 5-49 Internal structure of 32-bit descriptors	5-43



Tables

Table 5-1 DDR3 SDRAMs supported by the DDRC	5-2
Table 5-2 Command truth table of the DDRC	5-7
Table 5-3 Address mapping when the DDRC is in 32-bit mode.....	5-10
Table 5-4 Address mapping when the DDRC is in 16-bit mode.....	5-10
Table 5-5 K9F2G08U0M addresses	5-16
Table 5-6 K9GAG08X0M addresses	5-16
Table 5-7 Common commands for operating the NAND flash	5-16
Table 5-8 Load parameters of signal lines	5-28
Table 5-9 Mapping between transmission modes and voltages	5-34
Table 5-10 Timing parameters of the MMC interface	5-34
Table 5-11 Reference/Default values of MMC_CMD when a non-data transfer command is executed	5-38
Table 5-12 Reference/Default values of MMC_CMD when a single data block or multiple data blocks are read	5-40
Table 5-13 Reference/Default values of MMC_CMD when a single data block or multiple data blocks are written	5-41
Table 5-14 Definition of each bit of DES0	5-43
Table 5-15 Definition of each bit of DES1	5-44
Table 5-16 Definition of each bit of DES2	5-45
Table 5-17 Definition of each bit of DES3	5-45
Table 5-18 Reference configuration of MMC_CMDARG when the resume operation is performed	5-48



5 Memory Interfaces

5.1 DDRC

5.1.1 Overview

The DDR3 SDRAM controller (DDRC) controls access to the DDR3 SDRAM.

5.1.2 Features

The DDRC has the following features:

- One DDR3 SDRAM chip select (CS)
- Configurable DDR3 SDRAM data bus bit width (32 bits or 16 bits). When the 32-bit mode is used, the capacity of the upper 16-bit DDR3 SDRAM is the same as that of the lower 16-bit DDR3 SDRAM.
- Maximum 2 GB storage capacity
- Maximum 800 MHz DDR3 SDRAM working frequency and 1600 Mbit/s data rate
- Various DDR3 SDRAM low-power modes, such as the power-down and self-refresh modes
- Burst8 transfer mode for the DDR3 SDRAM



5.1.3 Function Description

5.1.3.1 Application Block Diagram

The DDRC enables the master devices of the SoC such as the CPU to access the external DDR3 SDRAM. It supports the DDR3 SDRAM that complies with the JEDEC (JESD79) standard after the timing parameter registers of the DDRC are configured by using the CPU. [Table 5-1](#) lists the mainstream DDR3 SDRAMs supported by the DDRC. The descriptions in [Table 5-1](#) are based on the working frequencies of DDR3 SDRAMs. The restrictions such as the capacity are not taken into consideration.

Table 5-1 DDR3 SDRAMs supported by the DDRC

Vendor	400 MHz	533 MHz	667 MHz	800 MHz	Remarks
JESD79 (DDR3 standard)	DDR3-800 DDR3-1066 DDR3-1333 DDR3-1600	DDR3-1066 DDR3-1333 DDR3-1600	DDR3-1333 DDR3-1600	DDR3-1600	1, 2

NOTE

The maximum working frequency of the DDR3 SDRAM supported by the DDRC is 800 MHz.

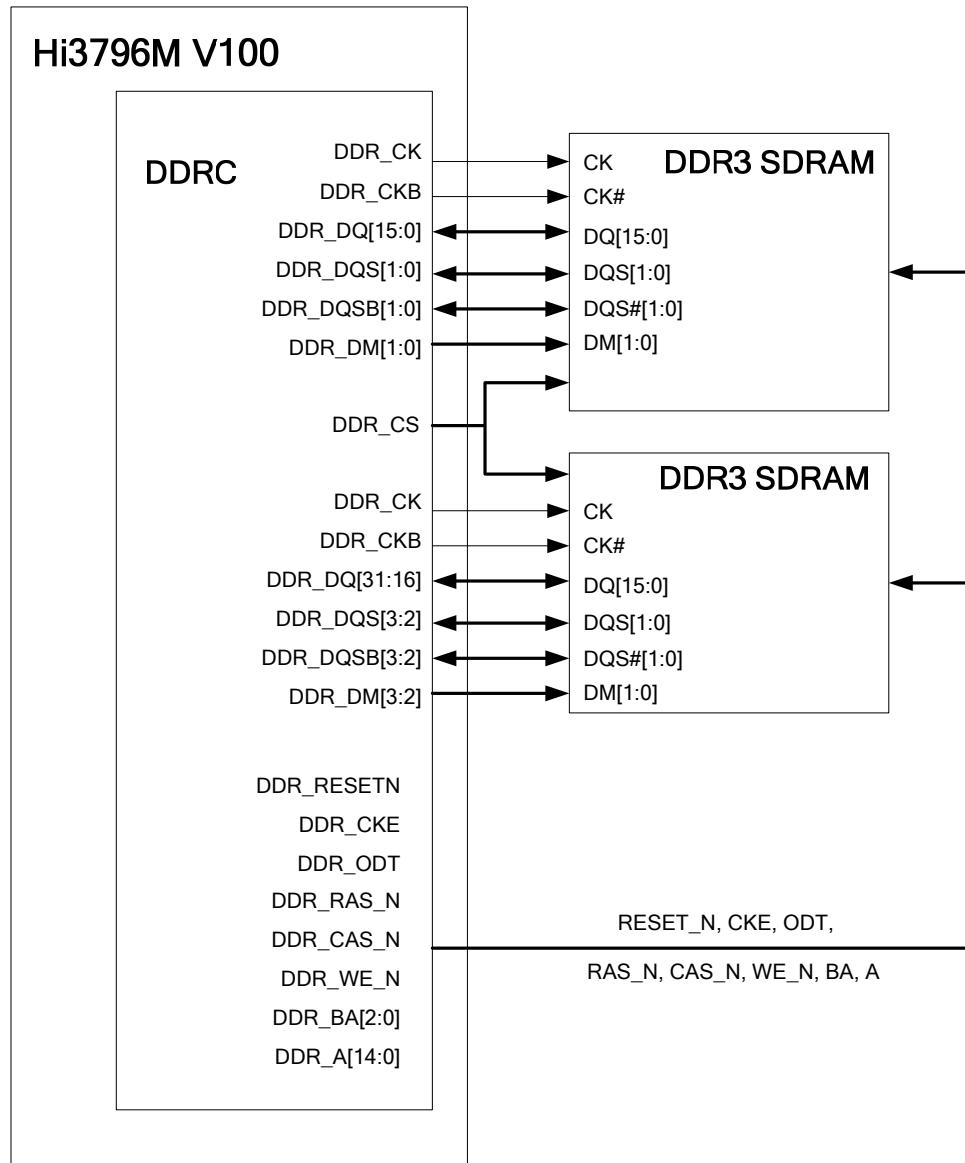
For details about the supported component types, see the JEDEC standard and the component data sheet.

The DDRC supports the 32-bit or 16-bit DDR. In 32-bit mode, the DDRC connects to two 16-bit DDR3 SDRAMs or four 8-bit DDR3 SDRAMs; in 16-bit mode, the DDRC connects to one 16-bit DDR3 SDRAM or two 8-bit DDR3 SDRAMs.

[Figure 5-1](#) and [Figure 5-2](#) show the connection between the DDRC and DDR3 SDRAMs in 32-bit mode.



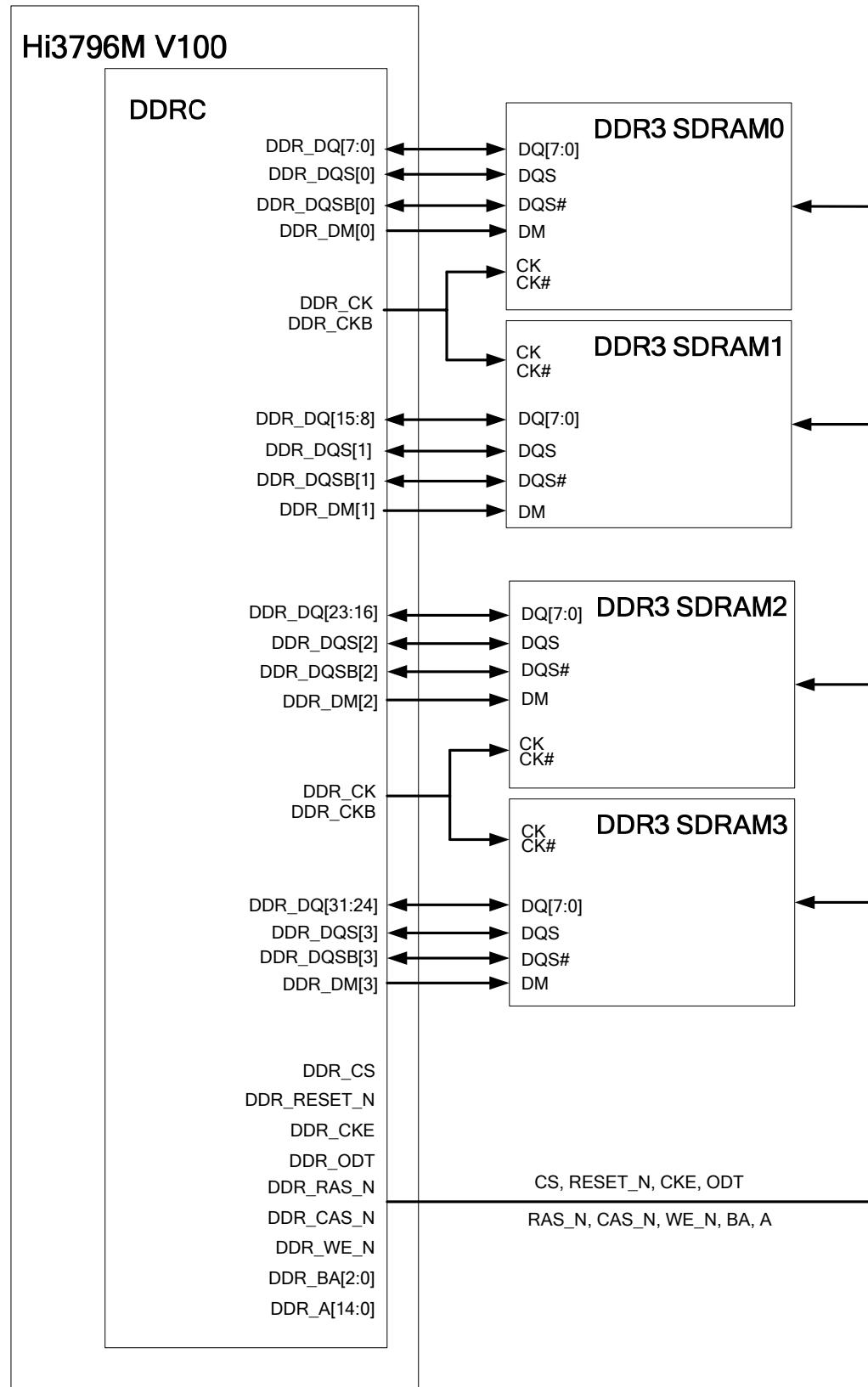
Figure 5-1 Connecting the DDRC to two 16-bit DDR3 SDRAMs in 32-bit mode



The DDR3 SDRAM is a 16-bit memory.

The DDRC command control

signals DDR_CS, DDR_CKE, DDR_RESET_N, DDR_RAS_N, DDR_CAS_N, DDR_WE_N, DDR_BA[2:0], DDR_A[14:0], and DDR_ODT connect to the command control signals of DDR3 SDRAMx.

Figure 5-2 Connecting the DDRC to four 8-bit DDR3 SDRAMs in 32-bit mode

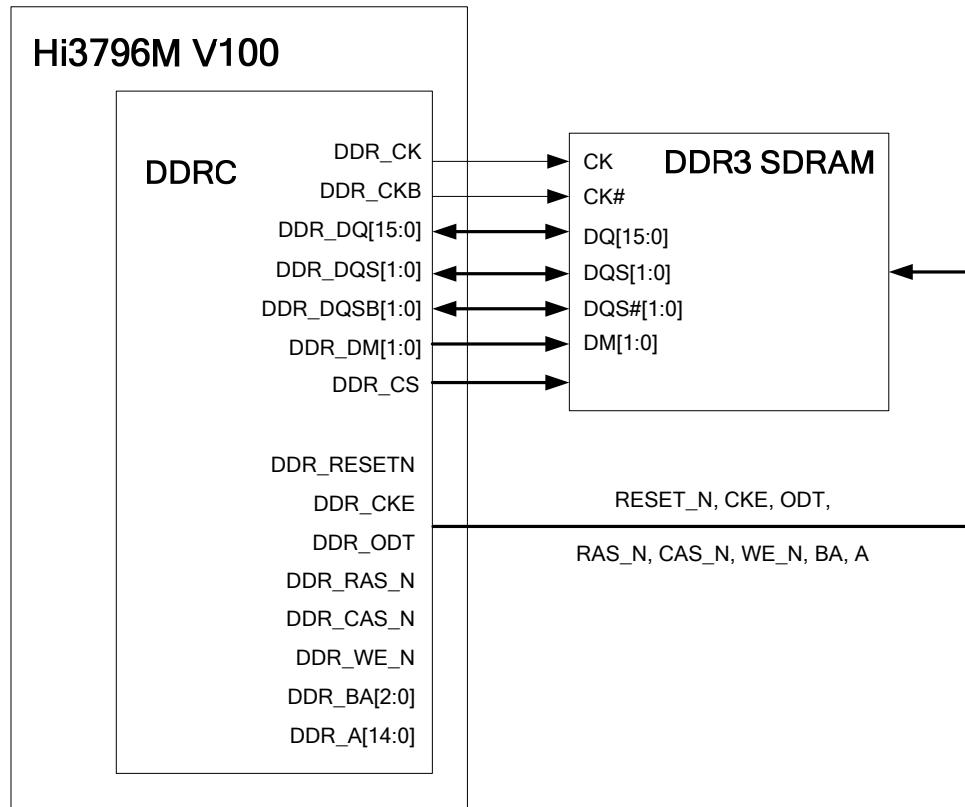
The DDR3 SDRAMx is an 8-bit memory.



The DDRC command control signals DDR_CS, DDR_CKE, DDR_RESET_N, DDR_RAS_N, DDR_CAS_N, DDR_WE_N, DDR_BA[2:0], and DDR_A[14:0] connect to the command control signals of DDR3 SDRAMx.

[Figure 5-3](#) and [Figure 5-4](#) show the connection between the DDRC and DDR3 SDRAMs in 16-bit mode.

Figure 5-3 Connecting the DDRC to a 16-bit DDR3 SDRAM in 16-bit mode

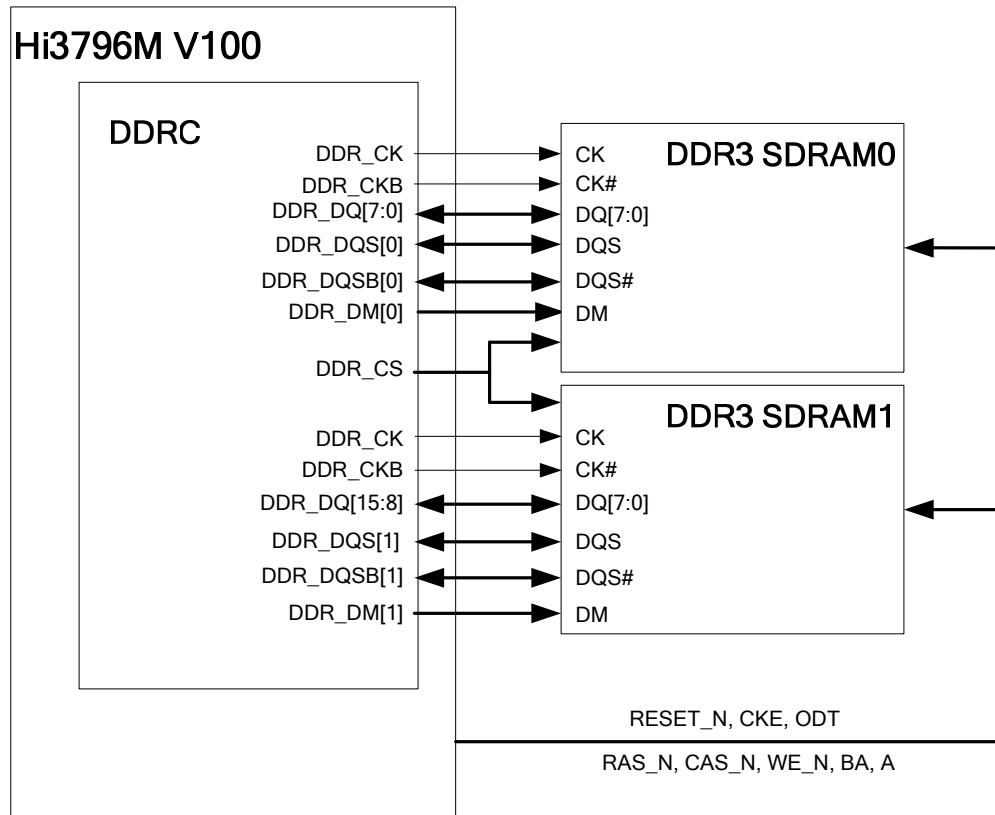


The DDR3 SDRAM is a 16-bit memory.

The DDRC command control signals DDR_CS, DDR_CKE, DDR_RESET_N, DDR_RAS_N, DDR_CAS_N, DDR_WE_N, DDR_BA[2:0], DDR_A[14:0], and DDR_ODT connect to the command control signals of the DDR3 SDRAM.



Figure 5-4 Connecting the DDRC to two 8-bit DDR3 SDRAMs in 16-bit mode



The DDR3 SDRAMx is an 8-bit memory.

The DDRC command control signals DDR_CS, DDR_CKE, DDR_RESET_N, DDR_RAS_N, DDR_CAS_N, DDR_WE_N, DDR_BA[2:0], and DDR_A[14:0] connect to the command control signals of DDR3 SDRAMx.

5.1.3.2 Function Implementation

As the timings of the DDRC interface comply with the JESD79 standard, the DDRC can access the data of the DDR3 SDRAM and control the status of the DDR3 SDRAM by sending the command words of the DDR3 SDRAM. For example, the DDRC can read and write to the DDR3 SDRAM, automatically refresh the DDR3 SDRAM, and control the power consumption of the DDR3 SDRAM.

Command Truth Table

The DDRC supports the read, write, and control command words of the DDR3 SDRAM. [Table 5-2](#) lists the command truth values of the DDRC. For details, see the JEDEC standard and component data sheet.

**Table 5-2** Command truth table of the DDRC

Function	DDR3_CKE	DDR3_CSN	DDR3_RAS_N	DDR3_CAS_N	DDR3_WEN	DDR3_ADDR			DDR3_BA
						15:11	AP(10)	9:0	
Deselect	H	H	X	X	X	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	V	L	V	BA
Precharge all	H	L	L	H	L	V	H	V	V
Auto refresh	H	L	L	L	H	V	V	V	V
Self refresh entry	H->L	L	L	L	H	V	V	V	V
Self refresh exit	L->H	L	H	H	H	V	V	V	V
Mode register set	H	L	L	L	L	V	V	V	V
ZQCL	H	L	H	H	L	X	H	X	X
ZQCS	H	L	H	H	L	X	L	X	X

H: high level; L: low level; V: valid; X: ignored.

ZQ calibration long (ZQCL): starts ZQ calibration on the DDR3 SDRAM when it is initialized during power-on.

ZQ calibration short (ZQCS): starts a calibration on the DDR3 SDRAM when its ambient environment is changed.

Auto Refresh

When DDRC_CFG_TIMING2 [taref] is set to a non-zero value, the DDRC refreshes the DDR3 SDRAM by automatically generating a periodical auto refresh command. At ambient temperature, the DDR3 SDRAM is required to implement 8,192 auto-refresh operations within 64 ms. That is, the auto-refresh cycle is 7.8 μ s. The relationship between the value of DDRC_CFG_TIMING2 [taref] (taref) and the auto-refresh cycle T (T = 7.8 μ s) is as follows:

$$T \geq taref \times (16 \times \text{DDR clock cycle})$$

When DDRC_CFG_TIMING2 [taref] is configured, the internal counter of the DDRC automatically loads the taref value and then counts in decremental 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 two modes of low-power management: common low-power mode and auto-refresh low-power mode.

When DDRC_CFG_PD [pd_en] is set to 1 to enable the SDRAM low-power mode, and the system is idle (the DDR is not read or written through the DDRC bus interface for a period), the DDR3 SDRAM enters the common low-power mode automatically.

When the system needs to enter standby mode, you can force the DDR3 SDRAM to enter the auto-refresh mode by setting DDRC_CTRL_SREF [sref_req] to 1. In this mode, the power consumption of the DDR3 SDRAM is minimized, data in the DDR3 SDRAM is retained, but the system cannot access the DDR3 SDRAM.

Arbitration Mechanism

The DDRC implements the priority scheduling algorithm and traffic control algorithm.

The priority mapping mode can be selected by configuring AXI_QOSCFG0n[id_map_mode]. Currently AXI_QOSCFG0n[id_map_mode] must be set to 1, that is, the system bus signal is used as the QoS priority. The DDRC adds priority attributes to the read/write commands sent to the bus by configuring AXI_WPRPRI and AXI_RDPRI, and then schedules the commands based on the priority attributes. In this way, the DDR3 SDRAM is accessed in a highly efficient manner.

The DDRC can also add the response delay attribute (timeout) to the commands sent to the bus by setting AXI_WRTOUT and AXI_RDTOUT to non-zero values. The mapping mode of the timeout period can be selected by configuring AXI_QOSCFG0n[tout_map_mode]. Currently AXI_QOSCFG0n[tout_map_mode] must be set to 1, that is, the 2-bit select signal of the system bus is used as the address to search for one of the four preset timeout periods. Then the bus command response timeout is ensured based on the response timeout priority scheduling algorithm.

By configuring DDRC_CFG_PERF[flux_en] and AXI_FLUX0n[flux_cfg0], the DDRC adds the traffic control attributes to bus interfaces and allocates traffic to each bus interface to ensure sufficient bandwidth of each interface when the traffic is heavy. The commands with the response timeout attributes are not restricted under traffic control after the timeout period elapses.

Traffic Statistics and Command Latency Statistics

The DDRC supports traffic statistics of the read/write operation based on an ID and the overall read and write traffic statistics. The DDRC can also collect statistics on the DDR interface usage. The statistic counter supports the consecutive count mode and one-shot count mode. In consecutive count mode, the counter is a non-saturation counter, and it is wrapped when the maximum count is reached. Therefore, the system needs to read the count value before the counter is wrapped. In one-shot count mode, the counter stops counting when the statistic time elapses. The system can obtain the instantaneous traffic and latency by using this function.

The DDRC supports command latency statistics, including the accumulative latency statistics.

The process for collecting statistics is as follows:

- Step 1** Set the statistic mode to consecutive triggering or one-shot triggering by configuring DDRC_CFG_PERF[perf_mode]. If one-shot triggering is selected, set the perf_prd field to configure the statistic cycle.



- Step 2** Set the ID for which statistics are to be collected by configuring DDRC_CFG_STAID.
- Step 3** Set the mask value of the ID by configuring DDRC_CFG_STAIDMSK. The DDRC determines whether to collect statistics on the current access based on sta_idmask and the accessed ID. Statistics of multiple IDs can be collected by using this method.
- Step 4** Enable statistics by configuring DDRC_CTRL_PERF[perf_en]. For the one-shot counting, counting is complete when DDRC_CTRL_PERF[perf_en] is restored to 0. For the consecutive counting, software needs to write 0 to DDRC_CTRL_PERF[perf_en] to stop counting.
- Step 5** Observe the statistics by using DDRC_HIS_FLUX_WR, DDRC_HIS_FLUX_RD, DDRC_HIS_FLUX_WCMD, DDR_C_HIS_FLUX_RCMD, DDRC_HIS_FLUXID_WR, DDRC_HIS_FLUXID_RD, DDRC_HI_S_FLUXID_WCMD, DDRC_HIS_FLUXID_RCMD, DDRC_HIS_WLATCNT0, DDRC_HI_S_WLATCNT1, DDRC_HIS_RLATCNT0, DDRC_HIS_RLATCNT1, and DDRC_HIS_INHERE_RLAT_CNT.

----End

Address Mapping Mode

The DDRC can convert the access address for the system bus into that for the DDR3 SDRAM. Set the address mapping mode to row-bank-column (RBC) by configuring DDRC_CFG_RNKVOL0 [mem_map]. Currently, only the RBC mode is supported. Then set the SDRAM row and column address bit widths by configuring DDRC_CFG_RNKVOL0[mem_row] and DDRC_CFG_RNKVOL0[mem_col] respectively. The DDRC then converts the system bus address into the DDR3 SDRAM address based on the address mapping algorithm.

The following describes the mapping algorithms for the system bus address and DDR3 SDRAM address by using the RBC mode as an example. Assume that the system bus address is BUSADR[31:0], the valid address is BUSADR[m – 1:0], and the address for the DDR3 SDRAM is DDRADDR[14:0]. When DDRADDR[14:0] serves as the row address, its valid address is DDRROW[x – 1:0]; when DDRADDR[13:0] serves as the column address, its valid address is DDCOL[y – 1:0]. In addition, the bank address for the DDR is DDRBA[z – 1:0] and the width of the storage data bus of the DDRC is DW. In this case, the address mappings are as follows:

- When DDRC_CFG_RNKVOL0[mem_map] is set to 2'b00, the RBC mapping mode is as follows:
$$\text{BUSADR}[m - 1:0] = \{\text{DDRROW}[x - 1:0], \text{DDRBA}[z - 1:0], \text{DDCOL}[y - 1:0], \text{DW}\{b0\}\}$$
In the preceding expression, the following condition is met: $m = x + y + z + DW$
 - When the DDRC is in 32-bit mode, the value of DW is 2.
 - When the DDRC is in 16-bit mode, the value of DW is 1.
- When DDRC_CFG_RNKVOL0[mem_map] is set to 2'b00 and A10 acts as the AP function bit of the DDR, the mapping between the system bus addresses and the DDR3 SDRAM addresses is described as follows.

Table 5-3 and Table 5-4 describe the address mappings in RBC mode.



Table 5-3 Address mapping when the DDRC is in 32-bit mode

Memory Type	Row Address Width	Column Address Width	DDR BA			Row/Column Address	DDR ADR								
			2	1	0		15	14	13	12	11	10/A P	9	8	7:0
1 Gbit 8 bank															
64 x 16	13	9	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	9	14	13	12	Row address	-	-	28	27	26	25	24	23	22:15
						Column address	-	-	-	-	-	AP	11	10	9:2
4Gbit 8bank															
256 x 16	15	10	13	12	11	Row address	-	29	28	27	26	25	24	23	22:15
						Column address	-	-	-	-	-	AP	11	10	9:2
8 Gbit 8 bank															
512 x 16	16	10	13	12	11	Row address	30	29	28	27	26	25	24	23	22:15
						Column address	-	-	-	-	-	AP	11	10	9:2

Table 5-4 Address mapping when the DDRC is in 16-bit mode

Memory Type	Row Address Width	Column Address Width	DDR BA			Row/Column Address	DDR ADR								
			2	1	0		13	12	11	10/A P	9	8	7:0		
256 Mbit 4 bank															
16 x 16	13	9	-	11	10	Row address	-	24	23	22	21	20	19:12		
						Column address	-	-	-	AP	-	9	8:1		
512 Mbit 4 bank															



Memory Type	Row Address Width	Column Address Width	DDR BA			Row/Column Address	DDR ADR						
			2	1	0		13	12	11	10/A P	9	8	7:0
32 x 16	13	10	-	12	11	Row address	-	25	24	23	22	21	20:13
						Column address	-	-	-	AP	10	9	8:1
1 Gbit 8 bank													
64 x 16	13	10	13	12	11	Row address		26	25	24	23	22	21:14
						Column address	-	-	-	AP	10	9	8:1
2 Gbit 8 bank													
128 x 16	14	10	13	12	11	Row address	27	26	25	24	23	22	21:14
						Column address	-	-	-	AP	10	9	8:1

5.2 NANDC

5.2.1 Overview

The NAND flash controller (NANDC) provides memory controller interfaces for connecting to external NAND flash memories to access data.

5.2.2 Features

The NANDC has the following features:

- Provides 9 KB (8192 bytes+1024 bytes) on-chip buffer for improving the read speed.
- Supports one CSs, one independent read_busy signals.
- Supports the CS don't care mode for reducing power consumption of the NAND flash.
- Supports bad block skipping and a maximum of 1 MB boot space.
- Supports boot without pin configuration. The component configuration information is automatically detected and adapted, including the page size, block size, and required error correcting code (ECC) capability.
- Supports the low-power mode.
- Supports NAND flash interfaces with 8-bit data bus.
- Allows the chip to boot from the NAND flash (with the page size of 2 KB, 4 KB, 8 KB, or 16 KB) corresponding to CS0.

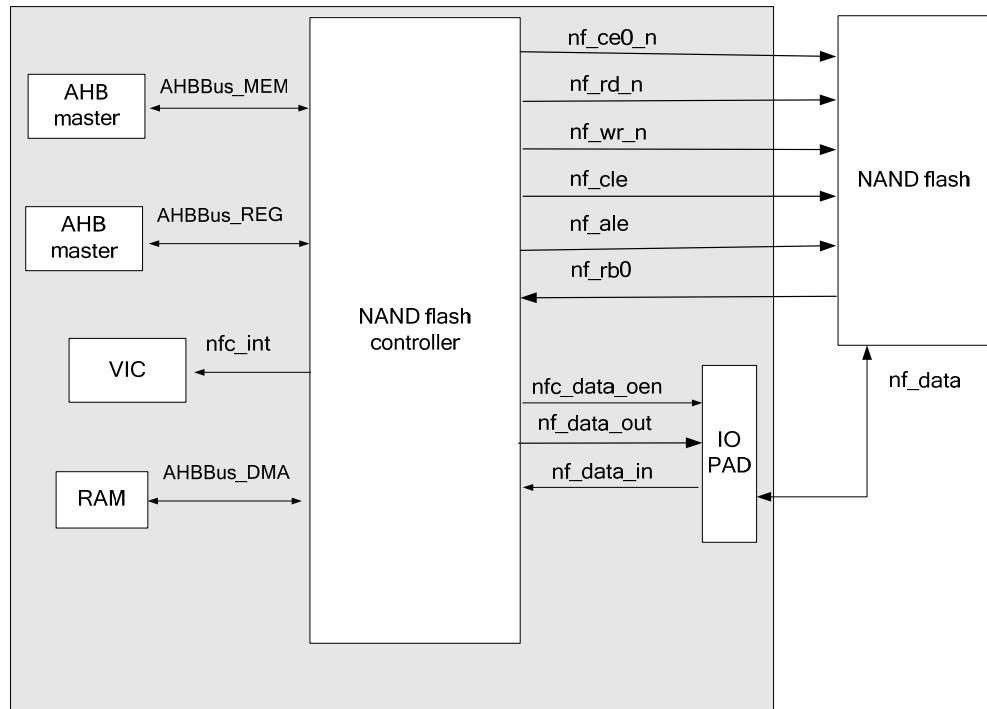


- Enables or disables ECC check and error correction.
 - 8 bits/1 KB, 16 bits/1 KB, 24 bits/1 KB BCH, and 28 bits/1 KB BCH code ECC check and error correction for the NAND flash with the 2 KB page size
 - 8 bits/1 KB, 16 bits/1 KB, 24 bits/1 KB, and 28 bits/1 KB BCH code ECC check and error correction for the NAND flash with the 4 KB page size
 - 24 bits/1 KB, 28 bits/1 KB, 40 bits/1 KB, and 64 bits/1 KB BCH code ECC check and error correction for the NAND flash with the 8 KB page size
 - 40 bits/1 KB and 64 bits/1 KB BCH code ECC check and error correction for the NAND flash with the 16 KB page size
- Reports the read/write interrupt, erase interrupt, programming completion interrupt, and ECC check error interrupt.
- Supports read and write operations on data with variable length in ECC0 mode; write operations on the entire page in non-ECC0 mode; and direct memory access (DMA) read operations on only the out-of-band (OOB) or the entire page in non-ECC0 mode
- Flexibly configures the commands and addresses issued by the controller to support various NAND flash command operations (including cache read and write)
- Provides write protection for the NAND flash and allows write-protected addresses to be configured.
- Supports the lock and lock-down modes and allows the flash lock and flash global lock to be enabled and disabled. By default, the flash lock and flash global lock are disabled.
- Allows the randomizer function to be enabled for the NAND flash with the 8 KB or 16 KB page size.

5.2.3 Function Description

5.2.3.1 Block Diagram

Hi3796M V100 provides one CS and one ready/busy signal. [Figure 5-5](#) shows the block diagram of NANDC interfaces.

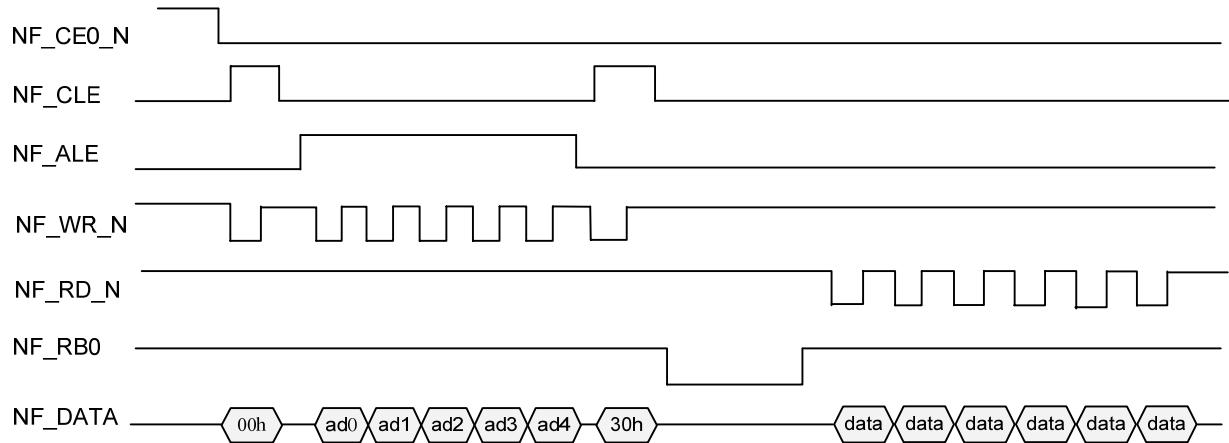
Figure 5-5 Block diagram of NANDC interfaces

5.2.3.2 Function Implementation

The data storage structure of the NAND flash consists of blocks and pages. Each block contains several pages. Before writing data to the NAND flash, you must erase the NAND flash by block, and then read/write to the NAND flash by page.

The commands for operating the NAND flash vary according to vendors. For details, see the appropriate NAND flash manual.

Figure 5-6 shows the typical timing when the NANDC reads a page of data from the NAND flash.

Figure 5-6 Typical timing when the NANDC reads a page of data from the NAND flash

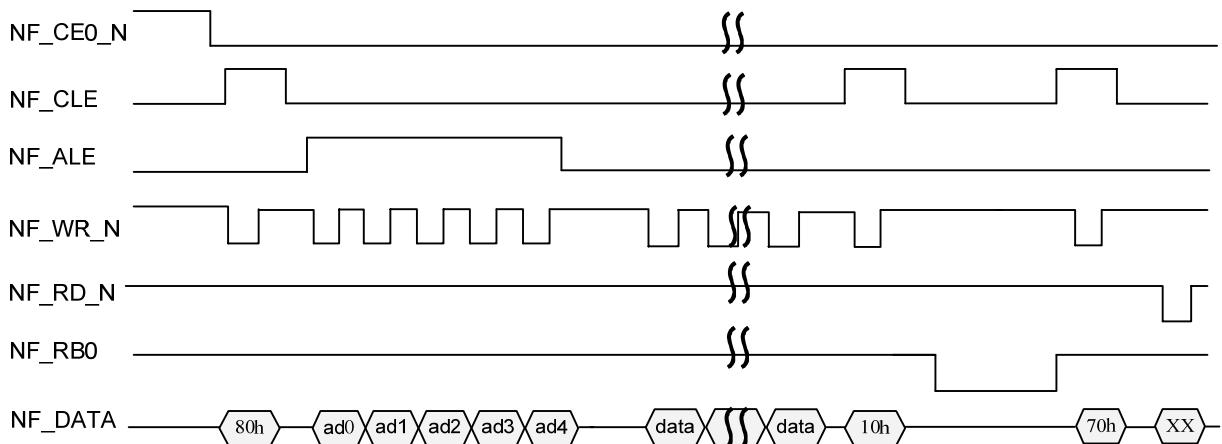
A typical read operation is performed as follows:

- Step 1** Send the read command 0x00 to the NAND flash.
- Step 2** Send the read start address. The start address consists of the page internal address, page address, and block address. For details, see the appropriate NAND flash manual.
- Step 3** Send the read confirmation command 0x30. The NAND flash pulls the RB signal low, indicating that the NAND flash is being read. About 25 μ s later, the RB signal becomes high, indicating that data of the NAND flash has been prepared.
- Step 4** Output the NF_RD_N signal, and sample the NAND flash data at the rising edge of the NF_RD_N signal.

----End

[Figure 5-7](#) shows the typical timing when the NANDC programs the NAND flash.

Figure 5-7 Typical timing when the NANDC programs the NAND flash



A typical program (write) operation is performed as follows:

- Step 1** Send the programming command 0x80 to the NAND flash.
- Step 2** Send the start address from which data is written. The start address consists of the page internal address, page address, and block address. For details, see the appropriate NAND flash manuals.
- Step 3** Write data to the internal buffer of the NAND flash.
- Step 4** Send the program confirmation command 0x10. The NAND flash pulls the RB signal low, indicating that the NAND flash is being programmed. About 200 ms later, the RB signal becomes high, indicating that the program operation is complete.
- Step 5** Send the command 0x70 to read the NAND flash status to check whether the program operation is successful.

----End



5.2.3.3 Operating Mode

Boot Mode

- The NANDC is in boot mode by default. The system can boot from only the NAND flash corresponding to CS0, even when the randomizer is enabled.
- The CPU can directly read data stored in a 1 MB address space in boot mode.
- When the system boots from the NAND flash, the NANDC automatically sends a command to read the corresponding page of the NAND flash based on the address read by the CPU and then returns the corresponding data.
- The CPU cannot write to the internal buffer.
- Bad blocks can be automatically identified and skipped. If seven consecutive bad blocks are detected, the boot fails.
- The boot configuration pins must be configured properly based on the models of the connected NAND flash.
- The boot mode can be automatically configured based on the component type.
- When the boot mode is automatically configured based on the component type, the following specifications are supported:
 - 8-bit ECC (2 KB/4 KB page size)
 - 24-bit ECC (2 KB/4 KB/8 KB page size)
 - 40-bit ECC (8 KB/16 KB page size)
 - 60-bit ECC (8 KB/16 KB page size).

The randomizer can be enabled or disabled in 24-bit and 40-bit ECC modes (8 KB page size). The randomizer is disabled when the page size is 2 KB or 4 KB but enabled in other modes by default. The block size can be automatically configured to 64 pages, 128 pages, 256 pages, and 512 pages.

Normal Mode

The NANDC can be switched to the normal mode by setting NFC_CON[op_mode] to 1. In this mode, the CPU can erase, program, and read the NAND flash.

NAND Flash Addresses

The NANDC does not translate addresses. It sends the values of low-bit and upper-bit address registers to the NAND flash directly based on the number of addresses configured by the command configuration register. Therefore, the software needs to translate the address for the CPU into the address for the NAND flash and then write the translated address to the address register. For details about the address configuration requirements for each flash memory, see the user manual of the corresponding NAND flash.

Table 5-5 lists the requirements for setting the addresses for the Samsung K9F2G08U0M memory. Its capacity is 256 MB x 8 bits and its page size is 2 KB. A0 to A11 indicate the page internal addresses (column addresses), and A12 to A27 indicate the page addresses (row addresses).



Table 5-5 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
4th cycle	A20	A21	A22	A23	A24	A25	A26	A27

Table 5-6 lists the requirements for setting the addresses for the Samsung K9GAG08X0M memory. Its capacity is 2 GB x 8 bits and its page size is 4 KB. A0 to A12 indicate the page internal addresses (column addresses), and A13 to A31 indicate the page addresses (row addresses).

Table 5-6 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

Operation Commands

The NAND flash vendors may provide certain advanced commands. **Table 5-7** lists the common commands for operating the NAND flash.

Table 5-7 Common commands for operating the NAND flash

Operation	1 st cycle	2 nd cycle	Remarks
READ	00H	30H	-
PROGRAM	80H	10H	-
BLOCK ERASE	60H	D0H	-
READ ID	90H	-	-
READ STATUS	70H	-	-
RESET	FFH	-	-



5.2.4 Data Structure

The size of the NANDC internal buffer is $(8192 + 1024)$ bytes. This section describes the storage structures of the data read from or written to the NAND flash in the NANDC buffer.

5.2.4.1 Non-ECC Mode

When no ECC check is required for the NAND flash:

- During write operations, the NANDC transparently writes the data in buffers to the NAND flash without processing.
- During read operations, the NANDC writes data read from the NAND flash to the internal buffers without processing.
- The number of bytes of the read or written data is specified by the NFC_DATA_NUM register. The NAND flash with the 2 KB, 4 KB, 8 KB, or 16 KB page size is supported.
- The non-ECC mode is used when the read ID, set feature, and get feature operations are performed.

5.2.4.2 8-Bit/1024-Byte ECC Mode

2KB Page Size

[Figure 5-8](#) shows the data structure of the driver in 8-bit ECC mode (2 KB page size).

Figure 5-8 Data structure of the driver in 8-bit ECC mode (2 KB page size)



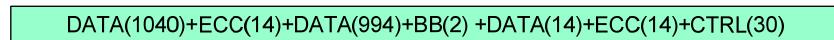
BB: bad block, 2 bytes

CTRL: control area for software, 30 bytes

The storage structure of the data in NANDC buffers is the same as that in the driver.

[Figure 5-9](#) shows the data structure of the data written to the NAND flash. The 2080-byte valid data of the software is divided into two 1040-byte data segments, and the 14-byte ECC is calculated for each data segment. Then the data segments and ECC code are written to the NAND flash alternately. Data structure of data in the NAND flash (with the 4 KB page size) in 8-bit ECC mode

Figure 5-9 Data structure of data in the NAND flash (with the 2 KB page size) in 8-bit ECC mode



4 KB Page Size

[Figure 5-10](#) shows the data structure of the driver in 8-bit ECC mode (4 KB page size).

Figure 5-10 Data structure of the driver in 8-bit ECC mode (4 KB page size)





BB: bad block, 2 bytes
CTRL: control area for software, 30 bytes

Figure 5-11 shows the data structure of the data written to the NAND flash. The software valid data is divided into four 1032-byte data segments, and the ECC is calculated for each data segment.

Data written to the NAND flash is automatically stored in three segments in the alternate form of 1032-byte data+14-byte ECC code. For the last data segment (1032-byte data+14-byte ECC code), the bad block flag is inserted in byte 4096 of the NAND flash.

Figure 5-11 Data structure of data in the NAND flash (with the 4 KB page size) in 8-bit ECC mode

DATA(1032)+ECC(14)+DATA(1032)+ECC(14)+DATA(1032)+ECC(14)+DATA(958)+BB(2) +DATA(42)+ECC(14)+CTRL(30)

5.2.4.3 16-Bit/1024-Byte ECC Mode

2 KB Page Size

Figure 5-12 shows the data structure of the driver in 16-bit ECC mode (2 KB page size).

Figure 5-12 Data structure of the driver in 16-bit ECC mode (2 KB page size)

DATA(2048B) BB CTRL RSV

BB: bad block, 2 bytes
CTRL: 6 bytes

The storage structure of the data in NANDC buffers is the same as that in the driver.

Figure 5-13 shows the data structure of data written to the NAND flash when the OOB length is 8 bytes. The 2056-byte valid data of the software is divided into two 1028-byte data segments, and the 28-byte ECC is calculated for each data segment. Then the data segments and ECC code are written to the NAND flash alternately.

Figure 5-13 Data structure of data in the NAND flash (with the 2 KB page size) in 16-bit ECC mode

DATA(1028)+ECC(28)+DATA(992)+BB(2)+DATA(28)+ECC(28)+CTRL(6)

4 KB Page Size

Figure 5-14 shows the data structure of the driver in 16-bit ECC mode (4 KB page size).

Figure 5-14 Data structure of the driver in 16-bit ECC mode (4 KB page size)

DATA(4096) BB CTRL

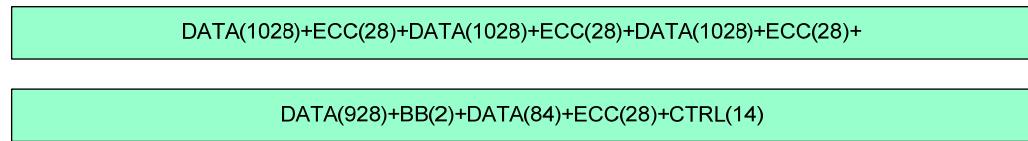


BB: bad blocks, 2 bytes
CTRL: control area for software, 14 bytes

Figure 5-15 shows the data structure of data written to the NAND flash when the OOB length is 16 bytes. The software valid data is divided into four 1032-byte data segments, and the ECC is calculated for each data segment.

Data written to the NAND flash is automatically stored as three data segments in the alternate form of 1032-byte data+28-byte ECC code. For the last data segment (1032-byte data+28-byte ECC code), the bad block flag is inserted in byte 4096 of the NAND flash.

Figure 5-15 Data structure of data in the NAND flash (with the 4 KB page size) in 16-bit ECC mode



5.2.4.4 24-Bit/1024-Byte ECC Mode

2 KB Page Size

Figure 5-16 shows the data structure of the driver in 24-bit ECC mode (2 KB page size).

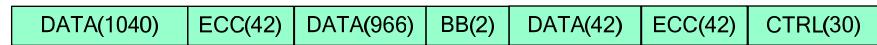
Figure 5-16 Data structure of the driver in 24-bit ECC mode (2 KB page size)



BB: bad block, 2 bytes
CTRL: control area for software, 30 bytes

Figure 5-17 shows the data structure of the data written to the NAND flash. The bad block is inserted in byte 2048 of the NAND flash.

Figure 5-17 Data structure of data in the NAND flash (with the 2 KB page size) in 24-bit ECC mode



4 KB Page Size

Figure 5-18 shows the data structure of the driver in 24-bit ECC mode (4 KB page size).

Figure 5-18 Data structure of the driver in 24-bit ECC mode (4 KB page size)





BB: bad blocks, 2 bytes
CTRL: control area for software, 30 bytes

The software valid data is divided into four 1024 bytes+8 bytes data segments, and a 42-byte ECC is calculated for each data segment.

When being written to the NAND flash, data and ECC code in the first three data segments are stored in the alternative form. For the last data segment (1024 bytes+8 bytes+42 bytes), the bad block flag is inserted in byte 4096 of the NAND flash.

Figure 5-19 Data structure of data in the NAND flash (with the 4 KB page size) in 24-bit ECC mode

DATA(1032)+ECC(42)+DATA(1032)+ECC(42)+DATA(1032)+ECC(42)+DATA(874)+BB(2)+DATA(126)+ECC(42)+CTRL(30)

8 KB Page Size

[Figure 5-20](#) shows the data structure of the driver in 24-bit ECC mode (8 KB page size).

Figure 5-20 Data structure of the driver in 24-bit ECC mode (8 KB page size)



BB: bad block, 2 bytes
CTRL: control area for software, 30 bytes

The software valid data is divided into eight 1024 bytes+4 bytes data segments, and a 42-byte ECC is calculated for each data segment. When being written to the NAND flash, the first seven data segments and ECC codes are stored alternately. For the last data segment (1024 bytes+4 bytes+42 bytes), the bad block flag is inserted in byte 8192 of the NAND flash.

Figure 5-21 Data structure of data in the NAND flash (with the 8 KB page size) in 24-bit ECC mode

DATA(1028)+ECC(42)+DATA(1028)+ECC(42)+DATA(1028)+ECC(42)+DATA(1028)+ECC(42)
DATA(1028)+ECC(42)+DATA(1028)+ECC(42)+DATA(1028)+ECC(42)+DATA(702)+BB(2)+DATA(294)+ECC(42)+CTRL(30)

5.2.4.5 28-Bit/1024-Byte ECC Mode

2 KB Page Size

[Figure 5-22](#) shows the data structure of the driver in 28-bit ECC mode (2 KB page size).

Figure 5-22 Data structure of the driver in 28-bit ECC mode (2 KB page size)



BB: bad blocks, 2 bytes

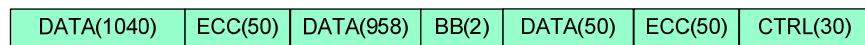


CTRL: control area for software, 30 bytes

The software valid data is divided into two 1024 bytes+16 bytes data segments, and a 50-byte ECC is calculated for each data segment.

When being written to the NAND flash, the first data segment and ECC code are stored alternately. For the last data segment (1024 bytes+16 bytes+50 bytes), the bad block flag is inserted in byte 2048 of the NAND flash.

Figure 5-23 Data structure of data in the NAND flash (with the 2 KB page size) in 28-bit ECC mode



4 KB Page Size

[Figure 5-24](#) shows the data structure of the driver in 28-bit ECC mode (4 KB page size).

Figure 5-24 Data structure of the driver in 28-bit ECC mode (4 KB page size)



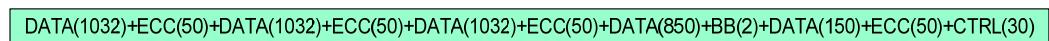
BB: bad blocks, 2 bytes

CTRL: control area for software, 30 bytes

The software valid data is divided into four 1024 bytes+8 bytes data segments, and a 50-byte ECC is calculated for each data segment.

When being written to the NAND flash, the first three data segments and ECC codes are stored alternately. For the last data segment (1024 bytes+8 bytes+50 bytes), the bad block flag is inserted in byte 4096 of the NAND flash.

Figure 5-25 Data structure of data in the NAND flash (with the 4 KB page size) in 28-bit ECC mode



8 KB Page Size

[Figure 5-26](#) shows the data structure of the driver in 28-bit ECC mode (8 KB page size).

Figure 5-26 Data structure of the driver in 28-bit ECC mode (8 KB page size)



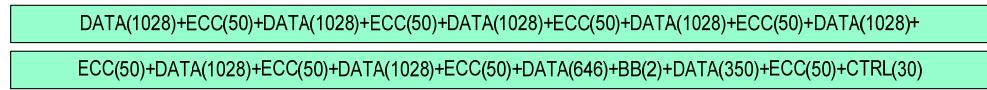
BB: bad blocks, 2 bytes

CTRL: control area for software, 30 bytes



The software valid data is divided into eight 1024 bytes+4 bytes data segments, and a 50-byte ECC is calculated for each data segment. When being written to the NAND flash, the first seven data segments and ECC codes are stored alternately. For the last data segment (1024 bytes+4 bytes+50 bytes), the bad block flag is inserted in byte 8192 of the NAND flash.

Figure 5-27 Data structure of data in the NAND flash (with the 8 KB page size) in 28-bit ECC mode



5.2.4.6 40-Bit/1024-Byte ECC Mode

8 KB Page Size

[Figure 5-28](#) shows the data structure of the driver in 40-bit ECC mode (8 KB page size).

Figure 5-28 Data structure of the driver in 40-bit ECC mode (8 KB page size)

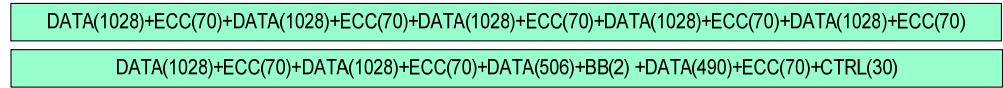


BB: bad blocks, 2 bytes

CTRL: control area for software, 30 bytes

The software valid data is divided into eight 1024 bytes+4 bytes data segments, and a 70-byte ECC is calculated for each data segment. When being written to the NAND flash, the first seven data segments and ECC codes are stored alternately. For the last data segment (1024 bytes+4 bytes+70 bytes), the bad block flag is inserted in byte 8192 of the NAND flash.

Figure 5-29 Data structure of data in the NAND flash (with the 8 KB page size) in 40-bit ECC mode



16 KB Page Size

[Figure 5-30](#) shows the data structure of the driver in 40-bit ECC mode (16 KB page size).

Figure 5-30 Data structure of the driver in 40-bit ECC mode (16 KB page size)



BB: bad blocks, 2 bytes

CTRL: control area for software, 30 bytes



The software valid data is divided into sixteen 1024 bytes+2 bytes data segments, and a 70-byte ECC is calculated for each data segment. When being written to the NAND flash, the first 15 data segments and ECC codes are stored alternately. For the last data segment (1024 bytes+2 bytes+70 bytes), the bad block flag is inserted in byte 16384 of the NAND flash.

Figure 5-31 Data structure of data in the NAND flash (with the 16 KB page size) in 40-bit ECC mode

DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+

DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+

DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+ECC(70)+DATA(1026)+

ECC(14)+BB(2)+ECC(56) +DATA(994)+ECC(70)+CTRL(30)

Two NANDC operations are required to write to each page. Write the 8192 bytes+16 bytes valid data to the start position of address 0 in the NANDC buffer, and then start the write-only mode of the NANDC. The ECC code generated by the NANDC for each operation is stored in the start position of the 8192 bytes+128 bytes in the internal buffer.

Reading an entire page also requires two NANDC operations, which is similar to writing to the page.

5.2.4.7 64-Bit/1024-Byte ECC Mode

8 KB Page Size

[Figure 5-32](#) shows the data structure of the driver in 64-bit ECC mode (8 KB page size).

Figure 5-32 Data structure of the driver in 64-bit ECC mode (8 KB page size)

DATA(8192) BB CTRL

BB: bad blocks, 2 bytes

CTRL: control area for software, 30 bytes

The software valid data is divided into eight 1024 bytes+4 bytes data segments, and a 112-byte ECC is calculated for each data segment. When being written to the NAND flash, the first seven data segments and ECC codes are stored alternately. For the last data segment (1024 bytes+4 bytes+112 bytes), the bad block flag is inserted in byte 8192 of the NAND flash.

Figure 5-33 Data structure of data in the NAND flash (with the 8 KB page size) in 64-bit ECC mode

DATA(1028)+ECC(112)+DATA(1028)+ECC(112)+DATA(1028)+ECC(112)+DATA(1028)+ECC(112)+

DATA(1028)+ECC(112)+DATA(1028)+ECC(112)+DATA(212)+BB(2) +DATA(784)+ECC(112)+CTRL(30)



16 KB Page Size

[Figure 5-34](#) shows the data structure of the driver in 64-bit ECC mode (16 KB page size).

Figure 5-34 Data structure of the driver in 64-bit ECC mode (16 KB page size)

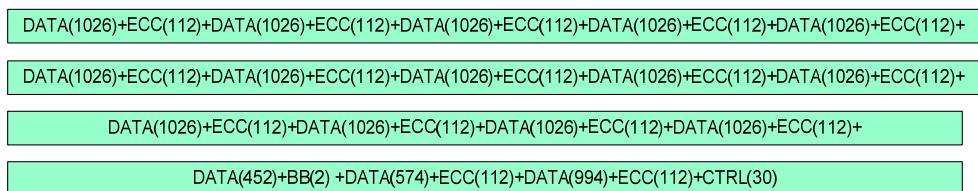


BB: bad blocks, 2 bytes

CTRL: control area for software, 30 bytes

The software valid data is divided into sixteen 1024 bytes+2 bytes data segments, and a 112-byte ECC is calculated for each data segment. When being written to the NAND flash, the first 14 data segments and ECC codes are stored alternately. For the fifteenth data segment (1024 bytes+2 bytes+112 bytes), the bad block flag is inserted in byte 16384 of the NAND flash.

Figure 5-35 Data structure of data in the NAND flash (with the 16 KB page size) in 64-bit ECC mode



Two NANDC operations are required to write to each page. Write the 8192 bytes+16 bytes valid data to the start position of address 0 in the NANDC buffer, and then start the write-only mode of the NANDC. The ECC code generated by the NANDC for each operation is stored in the start position of the 8192 bytes+128 bytes in the internal buffer.

Reading an entire page also requires two NANDC operations, which is similar to writing to the page.

5.2.5 Software Operations

5.2.5.1 Initializing the NAND Flash

The NAND flash is initialized as follows:

- Step 1** Write 1 to NFC_CON[op_mode] to enter the normal mode. Set NFC_CON[bus_width] and NFC_CON[page_size] based on the page size and bit width of the connected component, and write to NFC_CON[ecc_type] to set the ECC type.
- Step 2** Write to NFC_PWIDTH based on the timing requirements for the interconnected component.
- Step 3** In query mode, write to the interrupt enable register NFC_INTEN to mask all the interrupts; in interrupt mode, enable only the op_done interrupt and mask other interrupts.

----End



5.2.5.2 Erasing the NAND Flash

The NAND flash is erased as follows:

- Step 1** Write the address of the block to be erased to NFC_ADDRL and the erase command 0x0070_D060 to NFC_CMD.
- Step 2** Write 0x66d to NFC_OP to enable the NANDC to erase the NAND flash. (Assume that the NAND flash needs three addresses and the first NAND flash needs to be erased.)
- Step 3** In query mode, check whether the value of NFC_STATUS[nfc_ready] is 1. If yes, go to step 4; otherwise, continue the query. In interrupt mode, check whether the value of NFC_INTS[op_done] is 1. If yes, go to step 4.
- Step 4** Read NFC_STATUS[nf_status] to check whether the NAND flash is erased successfully.

----End

5.2.5.3 Writing to the NAND Flash in DMA Mode

- Configure the parameters of NFC_CON such as page size, ecc_type, and bus_wide based on the type of the interconnected component.
- Configure the base addresses for storing data in the DDR by setting NFC_BADDR_D0, NFC_BADDR_D1, NFC_BADDR_D2, NFC_BADDR_D3, and NFC_BADDR_OOB. Each base address stores 4 KB data (4-byte-aligned).
 - Set the base address register NFC_BADDR_D0 for the NAND flash with the 2 KB page size.
 - Set the base address register NFC_BADDR_D0 for the NAND flash with the 4 KB page size.
 - Set the base addresses registers NFC_BADDR_D0 and NFC_BADDR_D1 for the NAND flash with the 8 KB page size. Each base address stores 4 KB data.
 - Set the base address registers NFC_BADDR_D0, NFC_BADDR_D1, NFC_BADDR_D2, and NFC_BADDR_D3 for the NAND flash with the 16 KB page size. Each base address stores 4 KB data.
 - Set the base address register NFC_BADDR_OOB that specifies the storage address for the OOB data in the DDR for the NAND flash memory with the 2 KB, 4 KB, 8 KB, or 16 KB page size.
 - In ECC0 mode, configure NFC_DMA_LEN[len_oob] to set the size of the OOB area.
- Set NFC_DMA_CTRL to 0x73 to enable the NANDC to write to the NAND flash.
- The CPU waits until the NANDC sends the DMA transfer completion interrupt or queries NFC_DMA_CTRL[dma_start] until it is 0.

5.2.5.4 Reading the NAND Flash in DMA Mode

- Configure the parameters of NFC_CON such as page size, ecc_type, and bus_wide based on the type of the interconnected component.
- Configure the base addresses for storing the read data in the DDR by setting NFC_BADDR_D0, NFC_BADDR_D1, NFC_BADDR_D2, NFC_BADDR_D3, and NFC_BADDR_OOB. Each base address stores 4 KB data (4-byte-aligned).
 - Set the base address register NFC_BADDR_D0 for the NAND flash with the 2 KB page size.



- Set the base address register NFC_BADDR_D0 for the NAND flash with the 4 KB page size.
- Set the base addresses registers NFC_BADDR_D0 and NFC_BADDR_D1 for the NAND flash with the 8 KB page size. Each base address stores 4 KB data.
- Set the base address registers NFC_BADDR_D0, NFC_BADDR_D1, NFC_BADDR_D2, and NFC_BADDR_D3 for the NAND flash with the 16 KB page size. Each base address stores 4 KB data.
- Set the base address register NFC_BADDR_OOB that specifies the storage address for the OOB data in the DDR for the NAND flash memory with the 2 KB, 4 KB, 8 KB, or 16 KB page size.
- Set NFC_DMA_CTRL to 0x71 to enable the NANDC to read the NAND flash.
- If only the OOB is read in DMA mode, set NFC_DMA_CTRL to 0x1071 to enable the NANDC to read the NAND flash.
- The CPU waits until the NANDC sends the DMA transfer completion interrupt or queries NFC_DMA_CTRL[dma_start] until it is 0.

5.2.6 Others

Note the following:

- Reset the multi-level cell (MLC) NAND flash by following the memory manuals before using it.
- The operation commands of the NAND flash memories vary according to vendors. You need to configure the command register NFC_CMD according to memory manuals. In addition, the number of address cycles required by the NAND flash memories of different capacities is different. Therefore, you need to configure the address_cycles field of NFC_OP according to memory manuals. The timings supported by various memories are also different. You need to configure the read/write pulse width register NFC_PWIDTH and operation interval register NFC_OPIDLE according to memory manuals.
- After configuring relevant registers and buffer, write NFC_OP to enable the NANDC to read or write to the flash memory. After that, do not write to relevant registers; otherwise, the NANDC or NAND flash memory may be abnormal.
- After starting to read or write to the NAND flash by writing to NFC_OP, do not read or write to the buffer of the NANDC when NFC_STATUS[nfc_ready] is 0. Otherwise, error data may be returned.

5.3 MMC/SD/SDIO Controller

NOTE

Hi3796M V100 integrates two SDIO controllers. The SDIO0 controller supports the SDIO and SD cards, and the SDIO1 controller supports the eMMC and fSD.

5.3.1 Function Description

Functional Block Diagram

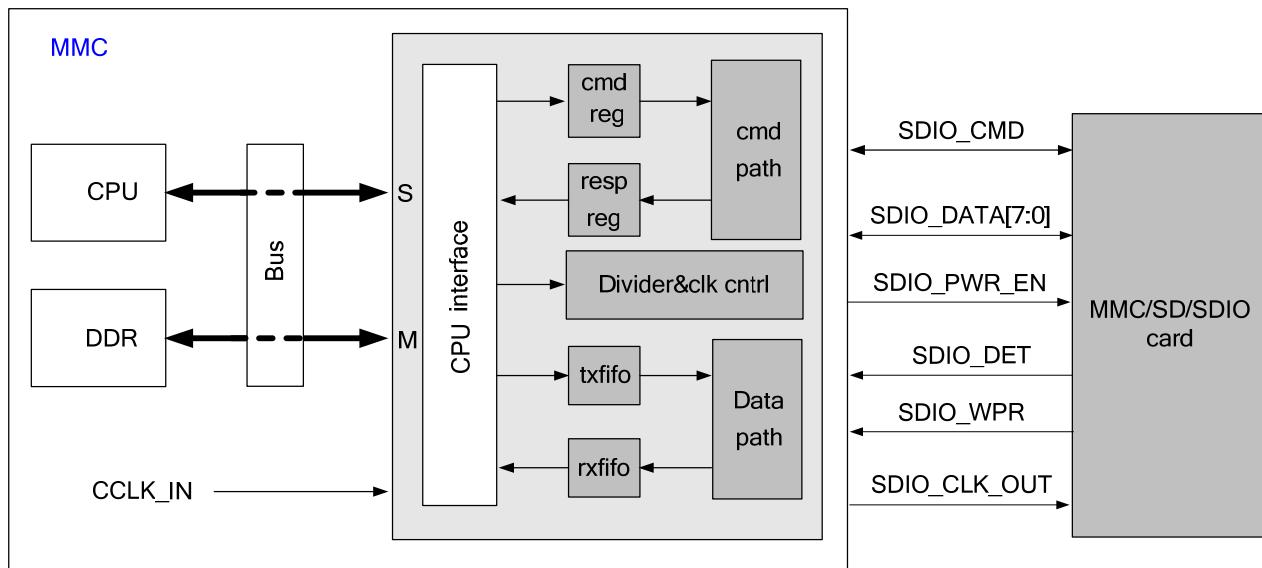
The MMC/SD/SDIO controller (MMC controller for short) controls the read/write operations on the MMC, supports various extended devices such as Bluetooth and Wi-Fi devices based on the SDIO protocol, and allows the system to boot from the MMC.

The MMC controller can control the devices that comply with the following protocols:

- SD mem-version 3.0
- SDIO-version 3.0
- MMC-version 4.41, eMMC 4.5, booting from the MMC

[Figure 5-36](#) shows the functional block diagram of the MMC controller.

Figure 5-36 Functional block diagram of the MMC controller



NOTE

S indicates the slave interface and M indicates the master interface.

The MMC controller connects to the system through the internal bus. It consists of the following parts:

- Command path
Transmits commands and receives responses.
- Data path
Reads and writes data by working with the command path.
- Interface clock control unit
Changes the frequency of the interface clock as required and enables or disables the interface clock. SDIO_CLK_OUT can be generated by dividing CCLK_IN.

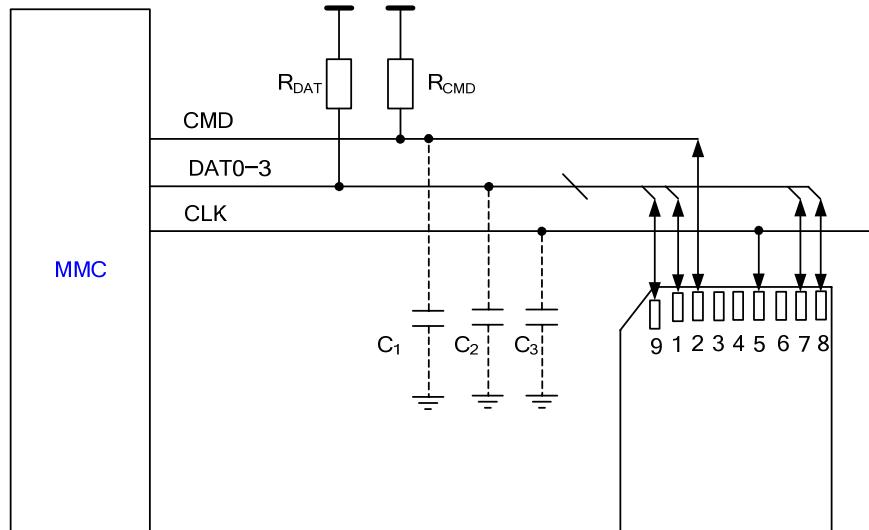
The MMC controller has the following features:

- Internal DMA data transmission
- One TX FIFO and one RX FIFO. Each FIFO is 256-word deep.
- Configurable burst size during DMA transmission and configurable FIFO threshold
- FIFO overflow and underflow interrupts to avoid errors during data transfer
- Cyclic redundancy check (CRC) generation and check for data and commands
- Programmable interface clock frequency
- MMC clock and interface clock disable in low-power mode
- 1-bit, 4-bit, or 8-bit data width based on the connected component
- Read/Write operations on data blocks with the size of 1 byte to 65,535 bytes
- Read/Write operations on stream data from/to the MMC card
- Suspend, resume, and read wait operations on the SDIO card

Typical Application

Figure 5-37 shows the typical application circuit of the MMC controller.

Figure 5-37 Typical application circuit of the MMC controller



The MMC controller exchanges commands and data with the connected card over a clock signal line, a bidirectional command signal line, and four or eight bidirectional data signal lines. The command signal line and data signal lines work in pull-up mode. Table 5-8 describes the pull-up resistors and the load capacitance of each signal line.

Table 5-8 Load parameters of signal lines

Parameter	Min	Max	Description
R_{DAT} , R_{CMD}	10 k Ω	100 k Ω	Pull-up resistors

Parameter	Min	Max	Description
Load capacitive reactance Cx	-	30 pF	$C_x = C_{mmchost} + C_{bus} + C_{card}$ The maximum load capacitance of each card (C_{card}) is 10 pF; therefore, the value of $C_{mmchost}$ plus C_{bus} must be less than 30 pF.
Inductive reactance	-	16 nH	$F_{pp} \leq 20 \text{ MHz}$



CAUTION

Besides the signal lines in [Figure 5-37](#), the card slot also provides the mechanical write protection signal and card detection signal. However, the signal interfaces are not provided in [Figure 5-37](#).

Commands and Responses

All interactions between the MMC controller and the card, including initializing the card, reading/writing to registers, querying the status, and transferring data, are implemented by using commands.

An MMC command is a segment of 48-bit serial data consisting of a start bit, a transmission bit, a command sequence, command parameters, a CRC bit, and an end bit. After receiving a command, the card returns a 48-bit or a 136-bit response based on the command type.

Figure 5-38 Format of an MMC command

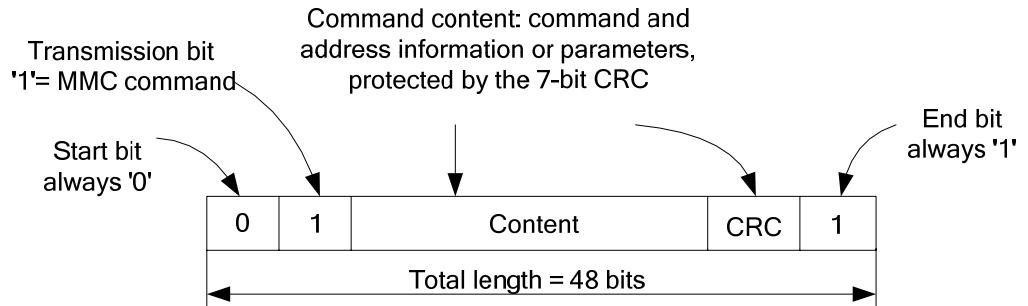
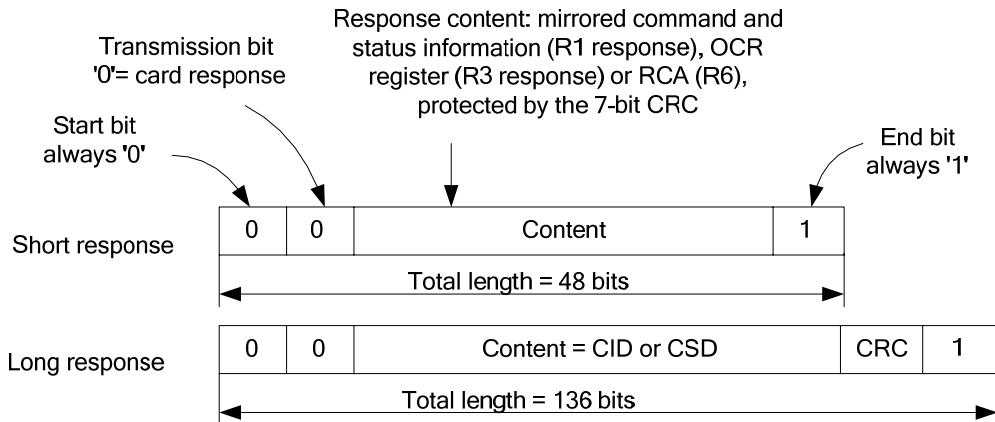
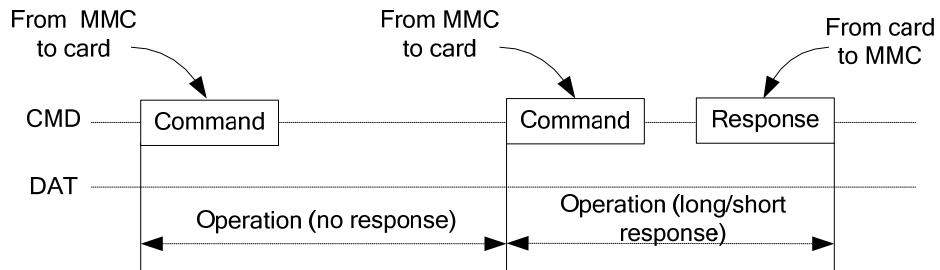


Figure 5-39 Format of the response to an MMC command

Commands are classified into the following two types based on whether data is transferred:

- Non-data transfer command
 - The MMC controller transmits commands/receives responses to/from the card through the command signal line CMD.
- Data transfer command
 - Besides the interaction on the command line, data is also transferred through the data lines DAT0 to DAT7.
 - 1. Non-data transfer command

Figure 5-40 shows the operations between the MMC controller and the card by running a non-data transfer command.

Figure 5-40 Running an MMC non-data transfer command

2. Data transfer command

The MMC supports the following data transfer commands:

- Stream data read/write command
 - Only the MMC card supports the stream data read/write command. In this case, only the data line DAT0 is used for data transfer, and no CRC check is performed.
- Single-block read/write command
 - After this command is sent, one single data block is read or written each time. No stop command is required for stopping each data transfer.



- Multi-block read/write command
 - Predefined block count mode

Before the multi-block read/write command is sent, the block count command is transmitted to specify the number of data blocks to be transferred.
 - Open ended mode

After a read/write command is sent, a stop command is required for stopping data transfer at the end of data transfer.

The difference between the two modes lies in how the MMC controller notifies the card of the end of each data transfer. The SD card supports only the open-ended mode, whereas the MMC supports both modes.

The multi-block read/write command for the SDIO card is different. To be specific, the command parameter contains the number of data blocks to be transferred when the read/write command is sent.

Commands are classified into the following three types based on responses:

- Command without response

For example, the card reset command.
- Command with short response

For example, the data transfer command and card status query command.
- Command with long response

This type of commands are used only to read the information about the card identification (CID) and card specific data (CSD) registers of a card.

Data Transmission

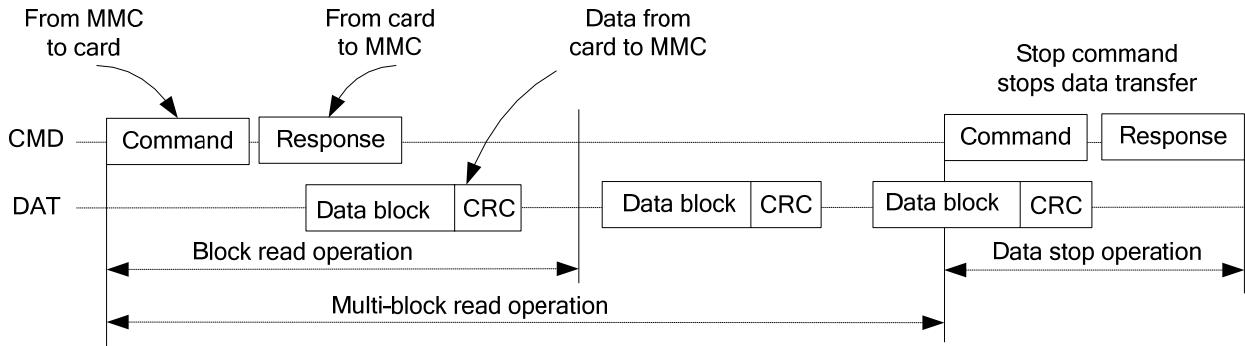
The single-block read/write command and the multi-block read/write command are widely used during data transfer. The data block during the data transfer of the SD card and MMC is 512 bytes; the block size during the data transfer of the SDIO card can be customized

NOTE

During the data transfer by using the block read/write command, the total amount of data to be transferred must be an integral multiple of the block size.

All data transfer commands are short response commands transferred with data through data lines. [Figure 5-41](#) and [Figure 5-42](#) show the relationship between the commands, responses, and timings of data lines.

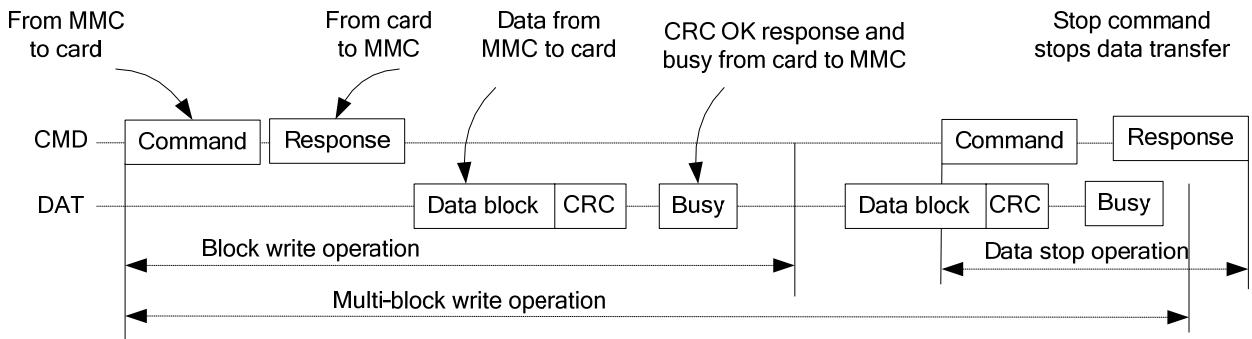
1. Single-block and multi-block read operations

Figure 5-41 Single-block and multi-block read operations

The MMC controller sends a single-block or multi-block read command to a card. When a response is being received, data is received by block. Each data block contains a CRC check bit to ensure the integrity of the transferred data.

In a single-block read operation, the data transfer is complete after the MMC controller receives a data block. In a multi-block read operation, the MMC controller needs to send a stop command to end the data transfer after receiving multiple data blocks (only in open-ended mode).

2. Single-block and multi-block write operations

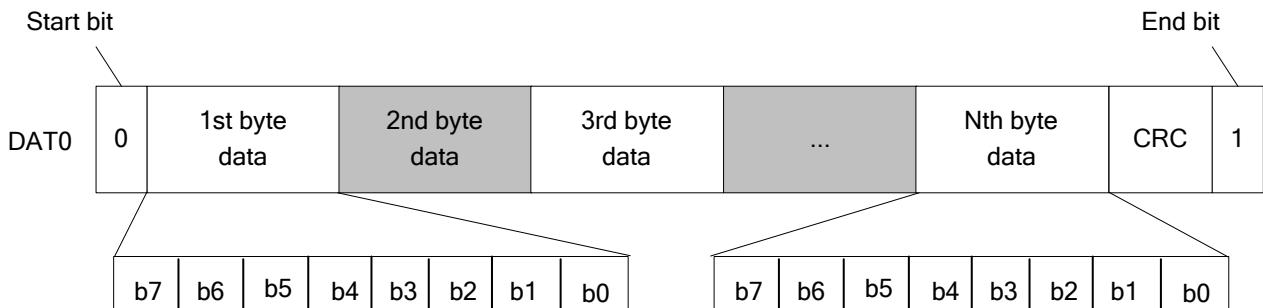
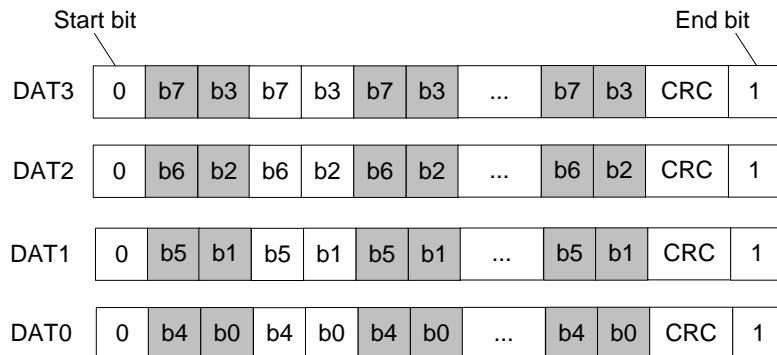
Figure 5-42 Single-block and multi-block write operations

The MMC controller sends a single-block or multi-block write command to a card. After receiving a response, the MMC controller starts to transmit data to the card by block. Each data block contains a CRC bit. The card performs CRC on each data block and returns the CRC status to the MMC controller to ensure proper data transfer.

In a single-block write operation, the data transfer is complete after the MMC controller transmits a data block. In a multi-block write operation, the MMC controller needs to send a stop command to end the data transfer after transmitting multiple data blocks (only in open-ended mode). After a write operation, the card may be busy in programming the flash memory. The MMC controller can perform the next operation on the card only after it confirms that the card is not busy by querying the status of the signal line DAT0.

3. Data transfer mode

- During the block read/write operations, the 1-/4-bit data line can be used to transfer data between the MMC controller and the card. Before sending a data transfer command, ensure that the data transfer widths of the MMC controller and card are the same (1-bit or 4-bit). You can set the data bit width of the MMC controller by configuring MMC_CTYPE and set the data bit width of the card by transmitting the corresponding command.
- [Figure 5-43](#) shows the data transfer format in 1-bit mode, and [Figure 5-44](#) shows the data transfer format in 4-bit mode.

Figure 5-43 Data transfer format in 1-bit mode**Figure 5-44** Data transfer format in 4-bit mode

Voltage Switching

The MMC controller supports SD 3.0 Ultra High Speed (UHS-1) and allows the voltage to be switched in SD mode. [Table 5-9](#) describes the mapping between the transmission modes and voltages.



Table 5-9 Mapping between transmission modes and voltages

Mode	Input Clock Source (MHz)	Maximum Output Clock (MHz)	Data Width	Maximum Bit Rate (MB/s)	Voltage (V)
SD_DS	15	15	1, 4	7.5	3.3
SD_HS	50	50	1, 4	25	3.3
SD_SDR12	15	15	1, 4	7.5	1.8
SD_SDR25	50	50	1, 4	25	1.8
SD_SDR50	75	75	1, 4	37.5	1.8
SD_SDR104	100	100	1, 4	50	1.8
SD_DDR50	50	50	1, 4	50	1.8
EMMC_DS	15	15	1, 4, 8	15	3.3/1.8
EMMC_HS	50	50	4, 8	50	3.3/1.8
EMMC_DDR	50	50	4, 8	100	3.3/1.8
EMMC_HS200	100	100	4, 8	100	1.8

5.3.2 Timings and Parameters

5.3.2.1 Timing Parameters

Table 5-10 describes the timing parameters of the MMC interface.

Table 5-10 Timing parameters of the MMC interface

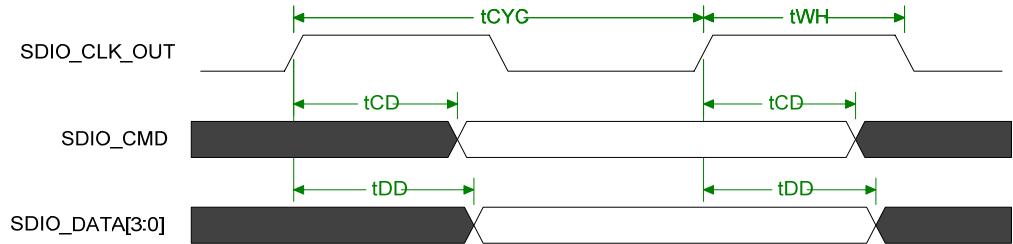
Parameter	Description	Min	Max	Unit
tCYC	Card clock cycle	20	-	ns
tWH	High level duration of the card clock	0.5tCYC	0.5tCYC	ns
tCCLK_IN	Working clock cycle of the MMC module	20 or 41.67		ns
tCD	SDIO_CMD output delay	0.6tCCLK_IN - 5.0	0.6tCCLK_IN + 1.2	ns
tDD	SDIO_DATA output delay	0.6tCCLK_IN - 4.7	0.6tCCLK_IN + 2.1	ns
tCS	SDIO_CMD input setup time	5.0	-	ns
tCH	SDIO_CMD input hold time	0.7	-	ns

Parameter	Description	Min	Max	Unit
tDS	SDIO_DATA input setup time	5.3	-	ns
tDH	SDIO_DATA input hold time	0.7	-	ns

5.3.2.2 Interface Timings

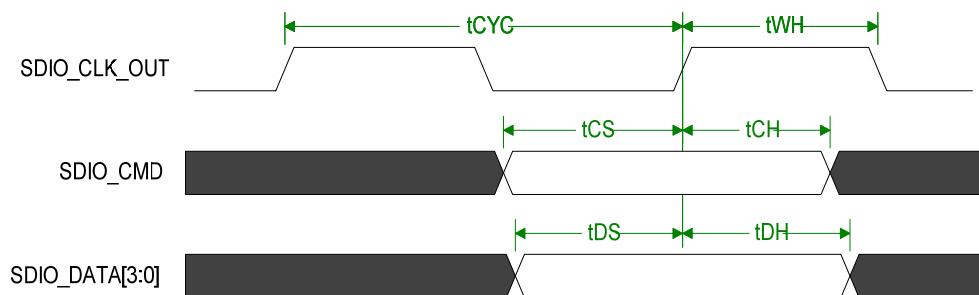
Output Timing

Figure 5-45 Output timing



Input Timing

Figure 5-46 Input timing



5.3.3 Application

Clock Gating

When the software completes the current command or data transfer and does not start a new data transfer, the SDIO_CLK_OUT clock can be disabled if the MMC controller is idle.

The process is as follows:



Step 1 Read the register MMC_STATUS.

Step 2 If both MMC_STATUS[Commandfsm_states] and MMC_STATUS[data_state_mc_busy] are 0, write 0 to MMC_CTRL to mask the MMC interrupt, enable the DMA request, and then go to step 3; if any of MMC_STATUS[Commandfsm_states] and MMC_STATUS[data_state_mc_busy] is not 0, wait until timeout, and then go to step 1.

Step 3 Write 0 to PERI_CRG40 bit[1] to disable the corresponding clock.

If you need to restart the working clock, write 1 to PERI_CRG40 bit[1].

----End

Soft Reset

If the MMC controller cannot restore to the idle state due to exceptions during data transfer, you can soft-reset the MMC controller by configuring PERI_CRG40 bit[4]. Then you can query MMC_STATUS[Data_busy] to check whether the MMC controller is idle.

Before using the MMC controller or after hot-plugging the card, you are advised to soft-reset the MMC controller.

Working Clock

Before using the MMC controller, you need to set the frequency of its working clock to 50 MHz by configuring PERI_CRG40 bit[9:8].

Interface Clock

The clock frequency varies according to the protocols that the MMC complies with and the state of MMC. The MMC controller provides an internal even frequency divider that generates appropriate interface clocks by dividing the frequency of the working clock. The following expression describes the relationship between the frequencies of the working clock CCLK_IN and the interface clock SDIO_CCLK_OUT of the MMC controller:

$$F_{SDIO_CCLK_OUT} = F_{CCLK_IN} / (2 \times \text{clk_divider})$$

Where, clk_divider is the value of MMC_CLKDIV[clk_divider]. The clock frequency varies according to card types. The maximum value of $F_{SDIO_CCLK_OUT}$ is 50 MHz.

Before changing the clock frequency of an MMC, ensure that no data or command is being transferred. To avoid glitches in the output clock to the MMC, you need to perform the following steps to change the clock frequency:

Step 1 Disable the interface clock.

Set MMC_CLKENA 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 Set the clock divider.

Configure MMC_CLKDIV based on 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 MMC_CLKENA 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 MMC_CLKDIV and MMC_CLKENA are loaded only after MMC_CMD[Start_cmd] and MMC_CMD[Update_clk_only] are set to 1. After the values are loaded successfully, the MMC controller clears MMC_CMD[Start_cmd] automatically. If another command is being executed, a hardware locked error (HLE) interrupt is generated. In this case, you need to clear the interrupt, and then send the command again.
- When a command is being executed or data is being transferred, the clock parameters of the card cannot be changed.

Initializing the MMC

Before commands and data are exchanged between a card and the MMC controller, you need to initialize the MMC controller by performing the following steps:

Step 1 Configure the working clock frequency of the MMC controller. For details, see "[Working Clock](#)."

Step 2 Soft-reset the MMC controller after the card is powered on and the command and data signal lines are pulled up and become stable. For details, see "[Soft Reset](#)."

Step 3 Clear interrupts. Set MMC_RINTSTS bit[15:0] to 1 to clear raw interrupt status bits.

Step 4 Set MMC_INTMASK bit[15:0] to 1 to enable all interrupt sources.

If data is transferred in DMA mode, set MMC_INTMASK bit[4] and MMC_INTMASK bit[5] to 0 to mask the TX/RX FIFO data request interrupts.

Step 5 Set MMC_CTRL[Int_enable] to 1 to enable the MMC interrupt.

Step 6 Configure the timeout parameter register MMC_TMOUT.

Step 7 Configure the FIFO parameter register MMC_FIFOTH.

----End

After the preceding steps, the interface clock can be configured and commands can be transmitted to the card.

Non-Data Transfer Command

If the MMC controller receives a response (correct response, error response, or timeout) after transmitting a command, it sets MMC_RINTSTS bit[2] to 1. Short responses are stored in MMC_RESP0, and long responses are stored in MMC_RESP0 to MMC_RESP3. MMC_RESP3 bit[31] is the most significant bit (MSB), and MMC_RESP0 bit[0] is the least



significant bit (LSB). After a command is transmitted, its error status is reflected by the response and the corresponding error bit of MMC_RINTSTS.

A non-data transfer command is transmitted as follows:

- Step 1** Configure the corresponding command parameters in MMC_CMDARG.
- Step 2** Configure the command register MMC_CMD according to section [5.3.3 "Application."](#)
- Step 3** Wait until the MMC controller runs the command. If the command is executed, the MMC controller clears MMC_CMD[Start_cmd] automatically.
- Step 4** Check whether an HLE interrupt is generated by MMC_RINTSTS bit[12].
- Step 5** Wait until the command is executed. If the MMC controller receives a response (correct response, error response, or timeout), it sets MMC_RINTSTS bit[2] to 1, indicating that the command is executed.
- Step 6** Check whether there is any response exception and read the response value if necessary.

You can check the response timeout, response CRC error, and response error by reading MMC_RINTSTS bit[8], MMC_RINTSTS bit[6], and MMC_RINTSTS bit[1] respectively.

----End



CAUTION

- The values of MMC_BYTCNT, MMC_BLKSIZ, MMC_CMDARG, and MMC_CMD are loaded only after MMC_CMD[Start_cmd] is set to 1 and MMC_CMD[Update_clock_registers_only] is set to 0. After the values are loaded successfully, the MMC controller clears MMC_CMD[Start_cmd] automatically.
- If other commands are being executed, an HLE interrupt is generated. In this case, you need to perform the preceding operations again. When a non-data transfer command is being executed, the values of MMC_BYTCNT and MMC_BLKSIZ are ignored.

Table 5-11 Reference/Default values of MMC_CMD when a non-data transfer command is executed

Parameter	Value	Description
Start_cmd	1	Command transfer start command
Update_clock_registers_onl y	0	Non-clock parameter update command
data_transfer_expected	0	Non-data transfer command
card_number	0	-
cmd_index	Cmd index	Command ID
send_initialization	0	This bit is set to 1 when the command is a card reset command such as CMD0.



Parameter	Value	Description
stop_abort_cmd	0	This bit is set to 1 when the command is a data transfer stop command such as 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, such as CMD0, CMD4, or CMD15.
Wait_prvdata_complete	1 or 0	The MMC controller must wait until the current data transfer command is executed before sending another command. It is recommended that this bit be fixed at 1 unless the command is used for querying the card status during data transfer or stopping the current data transfer.
Check_response_crc	1 or 0	This bit indicates whether the MMC controller checks the responded CRC bit.

Reading a Single Data Block or Multiple Data Blocks

To read a single data block or multiple data blocks, perform the following steps:

- Step 1** 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 number of bytes to be transmitted to MMC_BYTCNT.
- Step 3** Write the block size to MMC_BLKSIZ.
- Step 4** Write the start address for reading data to MMC_CMDARG.
- Step 5** Set the MMC_CMD register based on [Table 5-12](#).

For the SD card or MMC, you need to read a single data block by using CMD17, and read multiple data blocks by using CMD18; for the SDIO card, you can read a single data block or multiple data blocks by using CMD53.

The MMC controller starts to run commands after MMC_CMD is written. After commands are transferred to the bus, the cmd_done interrupt is generated.

- Step 6** Check the values of MMC_RINTSTS bit[5] and MMC_RINTSTS bit[10]. If any or both of them are 1, read the data in the FIFO from MMC_DATA. This ensures that the MMC controller can receive the subsequent data. In addition, check data error interrupts, that is, check the values of MMC_RINTSTS bit[7], MMC_RINTSTS bit[9], MMC_RINTSTS bit[13], and MMC_RINTSTS bit[15]. In this case, you can send a stop command to stop the data transfer by using the software.
- Step 7** If MMC_RINTSTS bit[3] is 1, data transfer is complete. In this case, read the remaining data in the FIFO by reading MMC_DATA.



If MMC_CMD[Send_auto_stop] is set to 1 when the command is executed, the MMC controller automatically transmits a stop command to stop the data transfer. For details, see "[Configuring the Auto-Stop Function.](#)"

----End

Table 5-12 Reference/Default values of MMC_CMD when a single data block or multiple data blocks are read

Parameter	Value	Description
Start_cmd	1	Command transfer start
Update_clock_registers_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 such as CMD0.
stop_abort_cmd	0	This bit is set to 1 when the command is a data transfer stop command such as CMD12.
send_auto_stop	0 or 1	For details, see " Configuring the Auto-Stop Function. "
transfer_mode	0	Block transfer
read/write	0	This bit indicates that data is read from the card.
rsponse_length	0	All responses to data commands are short responses.
data_transfer_expected	1	Data transfer command
rsponse_expect	1	This bit is set to 0 when a command is not responded such as CMD0, CMD4, or CMD15.
cmd_index	Cmd index	Command ID
Wait_prvdata_complete	1 or 0	The master device must wait until the current data transfer command is executed before transmitting another command. It is recommended that this bit be fixed at 1 unless the command is used for querying the card status or stopping the current data transfer.
Check_response_crc	1 or 0	This bit indicates whether the MMC controller checks the responded CRC bit.

Writing to a Single Data Block or Multiple Data Blocks

To write to a single data block or multiple data blocks, perform the following steps:



- 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 size of data to be transferred to MMC_BYTCNT.
- Step 3** Write the block size to MMC_BLKSIZ.
- Step 4** Write the start address for writing data to MMC_CMDARG.
- Step 5** Write data to the FIFO by writing to MMC_DATA. The FIFO needs to be filled with data completely at the very beginning.
- Step 6** Set the MMC_CMD register based on [Table 5-13](#).

For the SD card or MMC, you need to read a single data block by using CMD24, and read multiple data blocks by using CMD25; for the SDIO card, you can read a single data block or multiple data blocks by using CMD53.

- Step 7** Check the values of MMC_RINTSTS bit[4] and MMC_RINTSTS bit[10]. If any or both of them are 1, fill the FIFO with data by writing to MMC_DATA. In addition, check data error interrupts, that is, check the values of MMC_RINTSTS bit[7], MMC_RINTSTS bit[9], MMC_RINTSTS bit[13], and MMC_RINTSTS bit[15]. If necessary, the program can transmit a stop command to stop the data transfer. If MMC_RINTSTS bit[3] is 1, the data transfer is complete.

If MMC_CMD[Send_auto_stop] is set to 1 during command execution, the MMC controller automatically transmits a stop command to stop the data transfer. For details, see section "[Configuring the Auto-Stop Function](#)."

- Step 8** Query and wait until the value of MMC_STATUS[data_busy] is changed from 1 to 0.

----End

Table 5-13 Reference/Default values of MMC_CMD when a single data block or multiple data blocks are written

Parameter	Value	Description
Start_cmd	1	Command transfer start
Update_clock_registers_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 such as CMD0.
stop_abort_cmd	0	This bit is set to 1 when the command is a data transfer stop command such as CMD12.
send_auto_stop	0 or 1	For details, see section " Configuring the Auto-Stop Function ".
transfer_mode	0	Block transfer
read_write	1	This bit indicates that data is written to the card.
response_length	0	All responses to data commands are short responses.

Parameter	Value	Description
data_transfer_expected	1	Data transfer command
rspnse_expect	1	This bit is set to 0 when a command is not responded such as CMD0, CMD4, or CMD15.
cmd_index	Cmd index	-
Wait_prvdata_complete	1 or 0	The master device must wait until the current data transfer command is executed before transmitting another command. It is recommended that this bit be fixed at 1 unless the command is used for querying the card status or stopping the current data transfer.
Check_response_crc	1 or 0	This bit indicates whether the MMC controller checks the responded CRC bit.

Reading/Writing Stream Data

The modes of reading/writing stream data are the same as those of reading/writing data blocks, except that MMC_CMD[Transfer_mode] needs to be set to 1. During a stream data transfer, the auto-stop function is required.

Transferring Data by Using the IDMAC

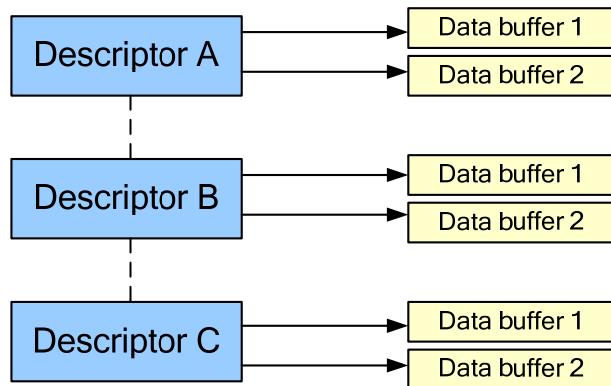
The MMC controller has an internal direct memory access controller (IDMAC) that transfers data from the original address to the destination address based on the specified descriptor.

Descriptors

The IDMAC supports the following two types of descriptors:

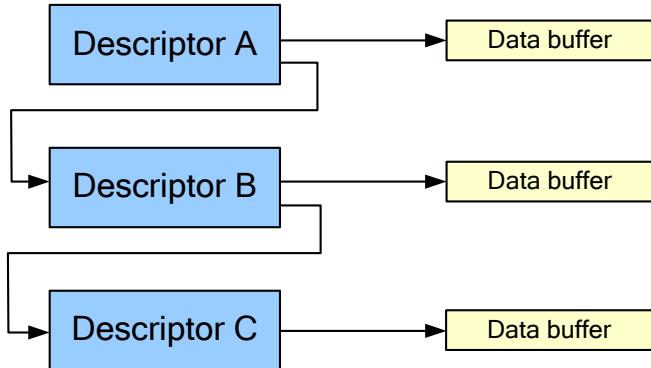
- Dual-buffer descriptor. For this type of descriptor, the span between two descriptors is determined by the DSL bit of MMC_BMOD. [Figure 5-47](#) shows the structure of the dual-buffer descriptor.

Figure 5-47 Structure of the dual-buffer descriptor



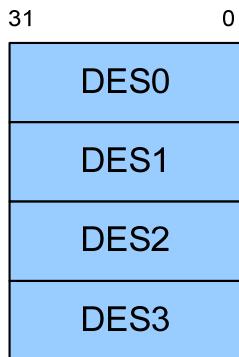
- Linked descriptor. For this type of descriptor, each descriptor points to only one buffer and the next descriptor. [Figure 5-48](#) shows the structure of the linked descriptor.

Figure 5-48 Structure of the linked descriptor



The descriptor must be word-aligned, and each descriptor contains 16-byte control and status information. [Figure 5-49](#) shows the internal structure of 32-bit descriptors.

Figure 5-49 Internal structure of 32-bit descriptors



DES0 is used to protect the control and status information. [Table 5-14](#) describes the definition of each bit.

Table 5-14 Definition of each bit of DES0

Bit	Name	Description
31	OWN	Descriptor attribute indicator 0: The descriptor belongs to the CPU. 1: The descriptor belongs to the IDMAC. After completing data transfer, the IDMAC clears this bit.



Bit	Name	Description
30	CES	Error status indicator when a card is read 0: No error occurs. 1: An error occurs.
29:6	RES	Reserved
5	ER	Descriptor link end indicator 0: The descriptor is not the last one on the link. 1: The descriptor is the last one on the link. This bit is valid only for the dual-buffer descriptor.
4	CH	Second address in DES3 0: address for the second buffer 1: address for the next descriptor When this bit is 1, DES1[25:13] must be 0.
3	FS	The descriptor contains the first data buffer when this bit is 1. If the size of the first data buffer is 0, the next descriptor contains the start data.
2	LD	The descriptor points to the last data buffer when this bit is 1.
1	DIC	The data transfer completion interrupt is masked when this bit is 1.
0	RES	Reserved

DES1 is used to specify the buffer size. [Table 5-15](#) describes the definition of each bit of DES1.

Table 5-15 Definition of each bit of DES1

Bit	Name	Description
31:26	RES	Reserved
25:13	BS2	Number of bytes in the second data buffer. This value must be an integral multiple of 4. This field is invalid when DES0[4] is 1.
12:0	BS1	Number of bytes in the first data buffer. This value must be an integral multiple of 4.

DES2 is the address pointer of the first data buffer. [Table 5-16](#) describes the definition of each bit of DES2.



Table 5-16 Definition of each bit of DES2

Bit	Name	Description
31:0	BAP1	Physical address for the first data buffer. The address must be word-aligned.

DES3 indicates the second address. [Table 5-17](#) describes the definition of each bit of DES3.

Table 5-17 Definition of each bit of DES3

Bit	Name	Description
31:0	BAP2	Physical address for the second data buffer when dual-buffer descriptors are used, or physical address for the next descriptor when DES0[4] is 1

Initializing the IDMAC

The IDMAC is initialized as follows:

- Step 1** Configure MMC_BMOD to set bus parameters.
 - Step 2** Configure MMC_IDINTEN to mask unnecessary registers.
 - Step 3** Create TX/RX descriptor linked lists, configure MMC_DBADDR, and set the start address.
 - Step 4** The IDMAC attempts to obtain descriptors from the descriptor linked lists.
- End

Transmitting Data

The data transmission procedure is as follows:

- Step 1** The CPU creates the descriptors DES0 to DES3, sets DES0 bit[31] (OWN bit) to 1, and prepares data buffers.
- Step 2** Write data commands to MMC_CMD.
- Step 3** Set TX_Wmark by using MMC_FIFOTH.
- Step 4** Obtain descriptors by using the IDMAC and check whether the value of the OWN bit is 1. If the OWN bit is not 1, wait until the CPU releases descriptors. During this process, the IDMAC enters the suspend state. Therefore, you need to configure MMC_PLDMND to enable the IDMAC to obtain descriptors again by using the CPU. If the OWN bit is 1, the IDMAC transfers data from data buffers to the internal FIFO of the MMC controller.

NOTE

If the interrupts are enabled, the corresponding bit of the IDMAC status register MMC_IDSTS is updated and the OWN bit is cleared after data transfer.

----End



Receiving Data

The data reception procedure is as follows:

- Step 1** The CPU creates the descriptors DES0 to DES3 and sets the DES0 bit[31] (OWN bit) to 1.
- Step 2** Write data read commands to MMC_CMD.
- Step 3** Set RX_Wmark by using MMC_FIFOTH.
- Step 4** Obtain descriptors by using the IDMAC and check whether the value of the OWN bit is 1. If the OWN bit is not 1, wait until the CPU releases descriptors. During this process, the IDMAC enters the suspend state. Therefore, you need to configure MMC_PLDMND to enable the IDMAC to obtain descriptors again by using the CPU.
- Step 5** If the OWN bit is 1, the IDMAC transfers data from the internal FIFO of the MMC controller to external data buffers.
- Step 6** If the interrupts are enabled, the corresponding bit of the IDMAC status register MMC_IDSTS is updated and the OWN bit is cleared after data transfer.

----End

Configuring the Auto-Stop Function

When multi-block data is read or written, a stop command is required to stop each data transfer. The stop command can be transmitted in non-data transfer command mode or by using the auto-stop function.

The auto-stop function is used in following scenarios:

- SD card
 - Multi-block read/write operation by using CMD18 or CMD25
- MMC card
 - Stream data read/write operation
 - Multi-block read/write operation in open-ended mode by using CMD18 or CMD25

You are advised to use the auto-stop function of the MMC controller. The configuration process is as follows:

- Step 1** Set MMC_CMD[Send_auto_stop] to 1 during execution of the block transfer command.
- Step 2** After data transfer is complete, the MMC controller automatically sends a stop command to enable the card to restore to the corresponding state.
- Step 3** Read MMC_RINTSTS[auto_cmd_done] to check whether the stop command is executed. The response is saved in MMC_RESP1.

----End

Stopping or Aborting Data Transfer

The stop command is used to end the data transfer between the MMC controller and the card, whereas the abort command is used to end the I/O data transfer only in SDIO_IOONLY or SDIO_COMBO mode.

The two commands are used as follows:



- Stop command

This command can be sent at any time during data transfer. Because this command is used to stop the data transfer, you need to set MMC_CMD bit[5:0] to CMD12, MMC_CMD bit[14] to 1, and MMC_CMD bit[13] to 0.

- Abort command

This command is available only for SDIO_IOONLY or SDIO_COMBO. To abort the data transfer, you need to configure the CCCR[ASx] register of the SDIO card by using CMD52.

Suspending and Resuming Data Transfer

The MMC controller can suspend the data transfer of a device with an SDIO card by performing a suspend operation. After a period of time, the MMC controller can resume the data transfer by performing a resume operation.

The suspend and resume operations are implemented by configuring the corresponding bits of the CCCR register of the SDIO card. The CCCR register is written or read by using 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.

If the BS bit is 1, 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 and the BR bits of the CCCR register are cleared.

The BS bit retains 1 when the data bus is being used. The BR bit retains 1 before the bus is released completely. When both the BR and BS bits are 0, the data transfer of the selected functional device is suspended.

Step 5 (Optional) If the current read operation is suspended, MMC_CTRL[Abort_read_data] needs to be set to 1 to reset the data transfer function of the MMC controller after the suspend operation succeeds. After reset, MMC_CTRL[Abort_read_data] 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, use the CMD7 command to select it. The card status can be queried by using the CMD52 or CMD53 command.

Step 3 Check whether the device to be resumed is ready for data transfer by querying the RF bit of the CCCR register. If the RF bit is 1, the device is ready for data transfer.

Step 4 Run the CMD52 command to write the device ID to the FS bit of the CCCR register to resume the data transfer, and enable the MMC controller to enter the data transfer state (that



is, write the block size to MMC_BLKSIZ, and write the amount of remaining data to be transferred to MMC_BYTCNT).

For details about the configuration of MMC_CMDARG, see [Table 5-18](#). The configuration of MMC_CMD is similar to the configuration during block transfer.

The data transfer is resumed after the CMD52 command is transmitted successfully. Read the resume data flag (DF) bit of the SDIO device. If this bit is 1, data transfer starts when the transfer function is resumed. If this bit is 0, no data needs to be transferred.

- Step 5** 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 5-18 Reference configuration of MMC_CMDARG when the resume operation is performed

MMC_CMDARG	Value	Description
Bit[31]	1	Read/Write flag
Bit[30:28]	0	ID of a functional device (by accessing the CCCR register)
Bit[27]	1	Real-time flag, that is, write-to-read
Bit[26]	-	-
Bit[25:9]	0x0D	Register address
Bit[8]	-	-
Bit[7:0]	ID of the functional device that is resumed	Write data



CAUTION

The MMC wakeup function is unavailable if the system enters low-power mode.

Performing the Read Wait Operation

The read wait operation is performed to suspend the data transfer of the device that is using the SDIO card. The MMC controller determines the duration of pausing the data transfer.

The read wait operation is performed as follows:

- Step 1** Check whether the card supports the read wait operation.

Read the SRW bit of the CCCR register by using the CMD52 command. If the SRW bit is 1, all the devices supporting the card support the read wait operation.

- Step 2** Set MMC_CTRL[Read_wait] to 1.



Step 3 If you want to resume the data transfer, clear MMC_CTRL[Read_wait].

----End



Contents

6 Data Steam Interfaces.....	6-1
6.1 ETH Ports.....	6-1
6.1.1 Overview.....	6-1
6.1.2 Function Description.....	6-1
6.2 TSI.....	6-2
6.2.1 Overview.....	6-2
6.2.2 Features.....	6-3



Figures

Figure 6-1 Logic block diagram of the ETH module 6-2



6 Data Steam Interfaces

6.1 ETH Ports

6.1.1 Overview

The ETH module provides an Ethernet MAC interface for receiving or transmitting data at 10 Mbit/s or 100 Mbit/s in half-duplex or full-duplex mode. This module also provides the media independent interface (MII) for the embedded PHY. With the eight configurable DMAC address filtering tables, the ETH module filters input frames received over the ETH port, limiting the traffic of the CPU port to protect the CPU against heavy traffic.

6.1.2 Function Description

The ETH module has the following features:

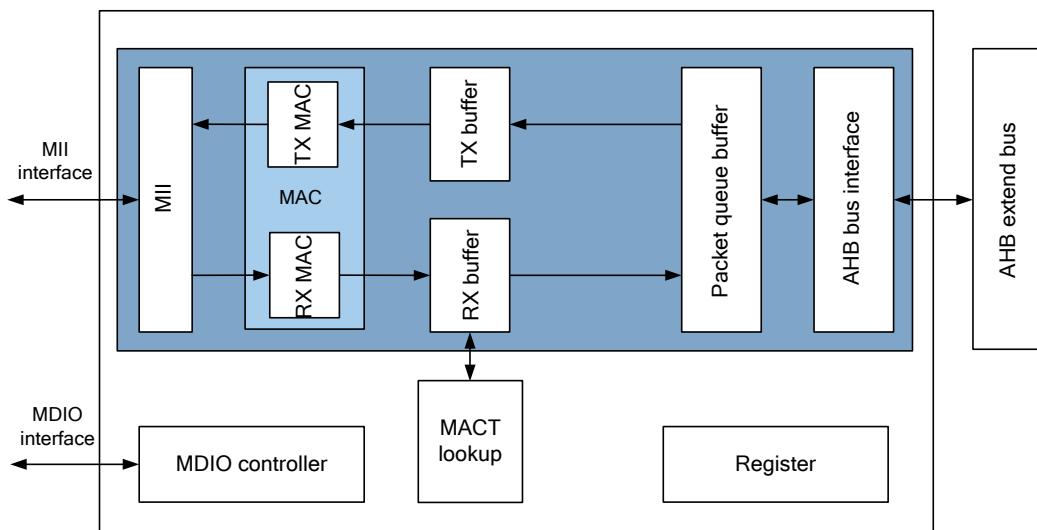
- One ETH MAC
- 10 Mbit/s or 100 Mbit/s speed
- Full-duplex or half-duplex mode
- MII for the embedded PHY
- Collision backoff and retransmission and late collision in half-duplex mode
- Pause frame transmission in full-duplex mode
- Frame length validity detection for discarding runt and giant frames
- Cyclic redundancy check (CRC) on received frames. Frames with CRC errors are discarded.
- CRC on frames to be transmitted
- Short frame filling
- Outloop in full-duplex mode
- MDIO interfaces with configurable clock frequencies
- Traffic limitation to prevent traffic attacks against the CPU
- Count of the received and transmitted frames
- Eight configurable DMAC address filtering tables
- Control over whether to receive or discard broadcast frames, multicast frames, and unicast frames based on configuration
- 802.3az EEE



- Wake-on-LAN (WoL)

Figure 6-1 shows the logic block diagram of the ETH module.

Figure 6-1 Logic block diagram of the ETH module



6.2 TSI

6.2.1 Overview

The TSI module is used to parse and demultiplex MPEG2 TSs complying with the standards of the ISO 13818-1 (GB 17975-1) system layer. It supports two DVB TS serial inputs, three internal memory TS inputs, and simultaneous processing of two TSs.



6.2.2 Features

The TSI module has the following features:

- Support for the MPEG2 TSs that comply with the ISO 13818-1 system layer standard
- Three internal memory TS inputs
- Maximum 360 Mbit/s processing rate
- Sync detection of 188-/204-byte TS packets
- 96 packet identifier (PID) channels
- DVB common scramble algorithm (CSA) 2.0 descrambling
- AES cipher block chaining (CBC) and electronic codebook (ECB) descrambling algorithms (residual blocks are not descrambled)
- AES IPTV CSA descrambling algorithm
- 3DES CBC and ECB descrambling algorithms (residual blocks are not descrambled)
- 32 groups of odd and even keys
- Advanced CA interfaces
- 96 groups of 16-byte filters
- TS recording
- Start code detect (SCD) extraction from eight channels
- Data reception in interrupt or query mode



Contents

7 Video Encoder.....	7-1
7.1 Overview	7-1
7.2 VEDU.....	7-1
7.2.1 Features.....	7-1
7.2.2 Function Description.....	7-2
7.3 JPGE.....	7-4
7.3.1 Features.....	7-4
7.3.2 Function Description.....	7-5



Figures

Figure 7-1 Function block diagram of the VEDU	7-3
Figure 7-2 Functional block diagram of the JPGE	7-5



7 Video Encoder

7.1 Overview

The video encoder consists of an embedded video encoding/decoding unit (VEDU) and video firmware (VFMW) running on the ARM processor, providing the maximum performance of 1080p@30 fps. With the features of low CPU usage, low bus bandwidth, short delay, and low power consumption, the VENC applies to various services such as video phone and video transcoding.

The JPEG encoder consists of an embedded hardware accelerator unit JPEG encoder (JPGE) and a VFMW. The JPGE can take snapshots with maximum 67 megapixels and encode Motion-JPEG (MJPEG) high-definition (HD) images.

7.2 VEDU

7.2.1 Features

The VEDU has the following features:

- ITU-T H.264 High Profile/Main Profile/Baseline Profile@L5 encoding
 - Motion prediction with 1/2 or 1/4 pixel precision
 - Inter-prediction of four subblock types (16x16, 16x8, 8x16, and 8x8)
 - Prediction modes of intra 4x4, intra 8x8, and intra 16x16
 - Deblocking filtering
 - Trans4x4 and trans 8x8
 - Context-based adaptive binary arithmetic coding (CABAC) and context-based adaptive variable-length coding (CAVLC) entropy encoding
 - I_PCM encoding
- Support for the following video input formats:
 - Semi-planar YCbCr4:2:0
 - Planar YCbCr4:2:0
 - Package YUYV4:2:2
- H.264 video encoding performance
 - 1x1080p@30 fps H.264 encoding



- 2x720p@30 fps H.264 encoding
- Configurable resolutions of input images
 - Minimum image resolution: 160 x 64
 - Maximum image resolution: 1920 x 2048
 - Image width or height step: 4
- Region of interest (ROI) encoding
 - Encoding of a maximum of eight ROIs
 - ROI encoding enable/disable control
- CBR mode and VBR mode
- Output bit rate ranging from 32 kbit/s to 20 Mbit/s
- Multi-channel encoding



CAUTION

Before video encoding, you can process images by using the video output unit (VOU). For example, you can implement deinterlacing, scaling, and icon overlapping.

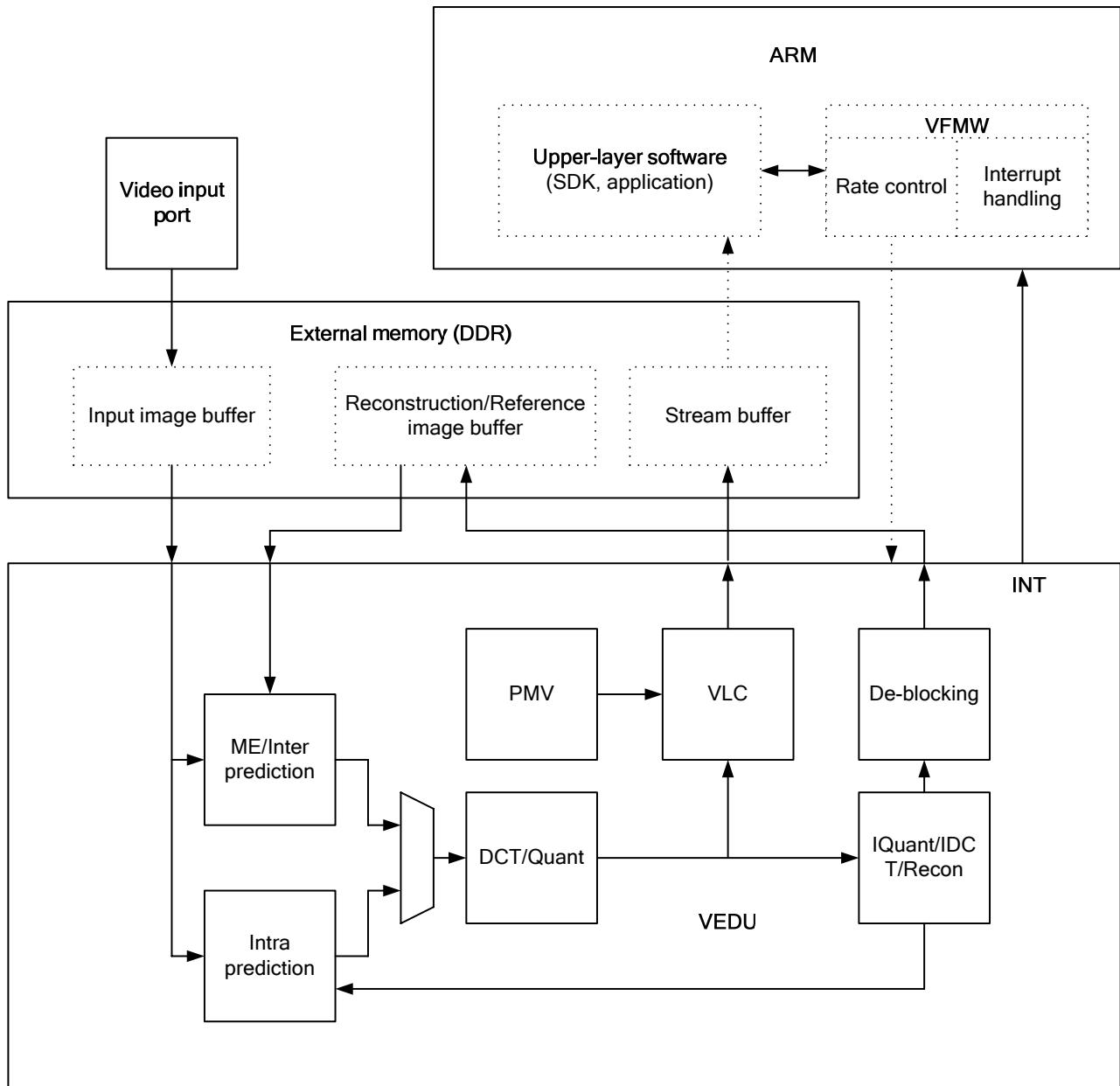
7.2.2 Function Description

Figure 7-1 illustrates the functions of the VEDU.

As shown in Figure 7-1, the VEDU hardware implements functions based on the protocols and algorithms that require a large amount of operands, such as motion estimation, inter-prediction, intra-prediction, motion vector prediction, transform/quantization, inverse transform/inverse quantization, variable length code (VLC) encoding, stream generation, and deblocking filtering. The VFMW controls the bit rate and handles interrupts.

Before the VEDU hardware is enabled for video encoding, the upper-layer software and VFMW allocate three types of buffers for the VEDU in the DDR SDRAM:

- Input image buffer: The VEDU reads the original images to be encoded from this buffer during encoding. This buffer is generally written by the video input unit.
- Reconstruction/Reference image buffer: The VEDU writes reconstruction images to the buffer during encoding. These reconstruction images are used as the reference images of subsequent images. When encoding P frames, the VEDU reads reference images from this buffer.
- Stream buffer: This buffer is used to store encoded streams. The VEDU writes streams to this buffer during encoding. This buffer is read by software.

Figure 7-1 Function block diagram of the VEDU

The VEDU performs encoding as follows:

- Step 1** The upper-layer software allocates an input image buffer and creates encoding channels.
- Step 2** The VFMW allocates a reconstruction image/reference image buffer and a stream buffer and creates contexts. During multi-channel encoding, the VFMW allocates a reconstruction image/reference image buffer, a stream buffer, and contexts for each encoding channel. Therefore, each encoding channel is independent of others.
- Step 3** The VFMW obtains the pointer of the input image buffer from the upper-layer software.
- Step 4** The VFMW configures the registers of the VEDU and enables the VEDU to encode a frame.
- Step 5** The VEDU reports an interrupt after encoding a frame.



- Step 6** The VFMW handles the interrupt, switches the buffer, adjusts the pointer, controls the bit rate, and then repeats step 3 for encoding the next frame.
- Step 7** The upper-layer software obtains streams from the VFMW.
- Step 8** The VFMW releases the reconstruction image/reference image buffer and the stream buffer and deletes contexts.
- Step 9** The upper-layer software destroys encoding channels.

----End

The VEDU can be soft-reset separately. For details, see section 3.2 "Reset."

7.3 JPGE

7.3.1 Features

The JPGE has the following features:

- ISO/IEC 10918-1 (CCITT T.81) baseline process (DCT sequential) encoding.
- Encoding of images that are sub-sampled in format of YCbCr4:2:0, YCbCr4:2:2, or YCbCr4:4:4
- Interleaved organization for the minimum coded unit (MCU)
- Support for multiple input image formats:
 - Planar YCbCr4:2:0
 - Planar YCbCr4:2:2
 - Planar YCbCr4:4:4
 - Semi-planar YCbCr4:2:0
 - Semi-planar YCbCr4:2:2
 - Package YUYV
- Maximum 1080p@30 fps encoding
- Configurable resolutions of input images
 - Minimum image resolution: 32 x 32
 - Maximum image resolution: 8192 x 8192
 - Image width or height step: 4
- Configurable quantization tables
 - An independent quantization table for the Y component, Cb component, and Cr component respectively



CAUTION

Before JPEG encoding, you can process images by using the VOU. For example, you can implement deinterlacing, scaling, and icon overlapping.

7.3.2 Function Description

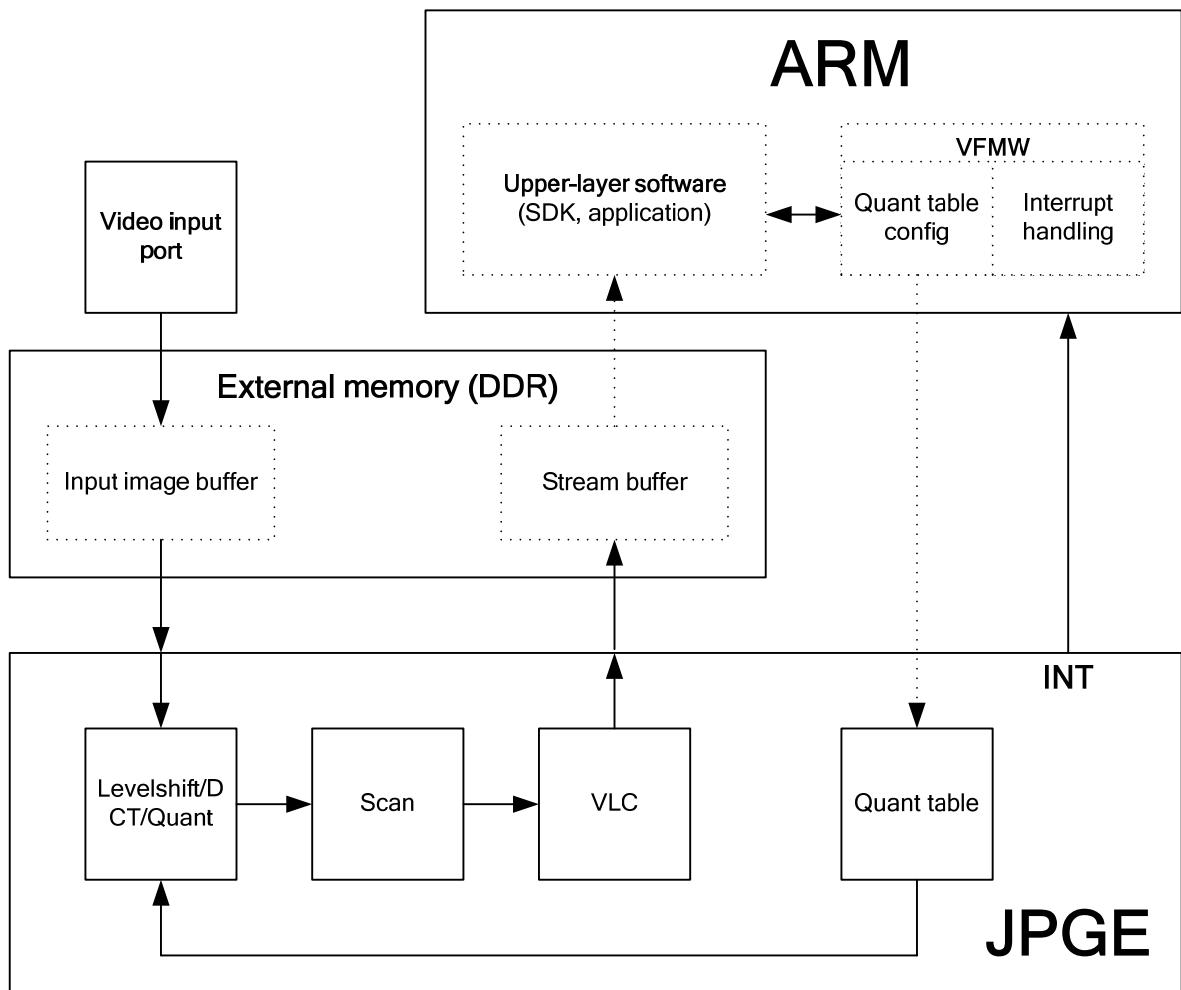
Figure 7-2 shows the functional block diagram of the JPGE.

Based on the protocols that require a large number of operands, the JPGE supports level shift, discrete cosine transform (DCT), quantization, scanning, VLC encoding, and stream generation. The VFMW configures quantization tables and handles interrupts.

Before the JPGE is enabled for video encoding, the upper-layer software and VFMW allocate two types of buffers for the JPGE in the DDR SDRAM:

- Input image buffer: The JPGE reads the original images to be encoded from this buffer during encoding. This buffer is generally written by the video input unit.
- Stream buffer: This buffer is used to store output encoded streams. The JPGE writes encoded streams to this buffer after encoding. This buffer is read by software.

Figure 7-2 Functional block diagram of the JPGE



The JPGE performs encoding as follows:

Step 1 The upper-layer software allocates an input image buffer and creates encoding channels.



- Step 2** The VFMW allocates a stream buffer and creates contexts. During multi-channel encoding, the VFMW allocates a stream buffer and contexts for each encoding channel. Therefore, each encoding channel is independent of others.
- Step 3** The VFMW obtains the pointer of the input image buffer from the upper-layer software.
- Step 4** The VFMW configures the registers of the JPGE and enables the JPGE to encode a frame.
- Step 5** The JPGE reports an interrupt after encoding a frame.
- Step 6** The VFMW handles the interrupt, switches the buffer, adjusts the pointer, and then repeats step 3 to encode the next frame (MJPEG) or exits (snapshots of JPEG images).
- Step 7** The upper-layer software obtains streams from the VFMW.
- Step 8** The VFMW releases the stream buffer and deletes contexts. Then encoding stops.
- Step 9** The upper-layer software destroys encoding channels.

----End

The JPGE can be soft-reset separately. For details, see section 3.2 "Reset."



Contents

8 Video Decoder	8-1
8.1 VDH	8-1
8.1.1 Overview	8-1
8.1.2 Features	8-1
8.1.3 Function Description	8-2
8.2 JPDG	8-3
8.2.1 Overview	8-3
8.2.2 Features	8-3
8.3 PGD	8-4
8.3.1 Overview	8-4
8.3.2 Features	8-4



Figures

Figure 8-1 Architecture of the video decoder 8-2



8 Video Decoder

8.1 VDH

8.1.1 Overview



CAUTION

Hi3796M V100 has an embedded video decoder that supports the H.264, H.265, MPEG1, MPEG2, MPEG4, AVS, VC1 (including WMV9), VP6, and VP8 protocols and provides superior video post-processing functions.

The video decoding module for high-definition (VDH) consists of the VFMW running on the ARM processor and an embedded video decoding engine. The VFMW obtains streams from the upper-layer software, parses the streams, and calls the video decoding engine to generate the decoding image sequences. Under the control of the upper-layer software, the video output unit (VOU) outputs the sequences to a monitor or other devices.

8.1.2 Features

The VDH has the following features:

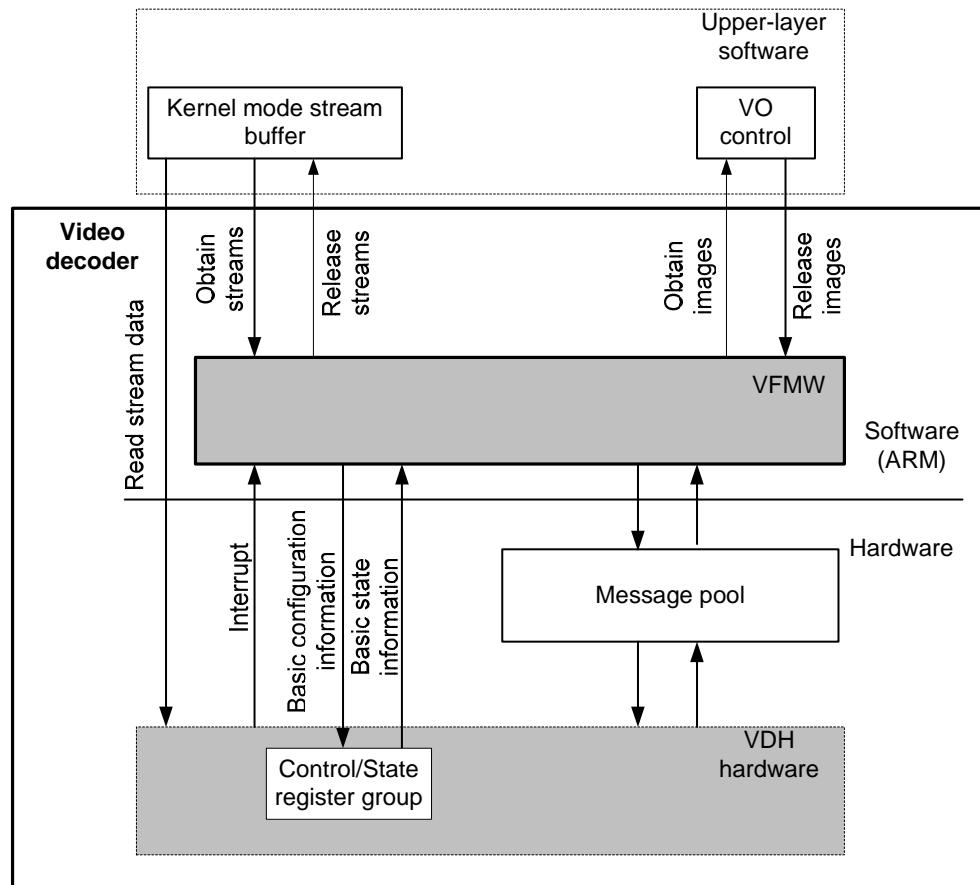
- ITU-T H.265 Main Profile@L5.0 high-tier (or lower levels)
 - Maximum 4K x 2K (3840 x 2160) video resolution
 - Maximum 100 Mbit/s or 4K x 2K@30 fps decoding rate
- ITU-T H.264 High Profile@L5.1 (or lower levels)
 - Maximum 4K x 2K (3840 x 2160) video resolution
 - Maximum 135 Mbit/s or 4K x 2K@30 fps decoding rate
- ISO/IEC 13818-2 (MPEG2) Main Profile@high level and backward-compatible with MP@ML, MP@LL, SP@ML, and ISO/IEC 11172-2 (MEPG1)
 - Maximum 1080p (1920 x 1080) video resolution
 - Maximum 80 Mbit/s or 60 fps decoding rate
- ISO/IEC 14496-2 (MPEG4) Advanced Simple Profile@L0–5, compatible with the short header format, ISO/IEC 14496-2 Simple Profile@L0–3, and DivX3/4/5/6

- Maximum 1080p (1920 x 1080) video resolution
- Maximum 50 Mbit/s or 60 fps decoding rate
- AVS baseline@L6.0
 - Maximum 1080p (1920 x 1080) video resolution
 - Maximum 50 Mbit/s or 60 fps decoding rate
- VC1 SP@ML, MP@HL, and AP@L3
 - Maximum 1080p (1920 x 1080) video resolution
 - Maximum 45 Mbit/s or 60 fps decoding rate
- VP6
 - Maximum 1080p (1920 x 1080) video resolution
 - Maximum 50 Mbit/s or 60 fps decoding rate
- VP8
 - Maximum 1080p (1920 x 1080) video resolution
 - Maximum 50 Mbit/s or 60 fps decoding rate

8.1.3 Function Description

Figure 8-1 shows the architecture of the video decoder.

Figure 8-1 Architecture of the video decoder





- VDH: a video decoding engine that supports multiple protocols
- VFMW: a software component running on the master processor for scheduling the VDH
- Message pool: a memory located in the external SDRAM for exchanging the messages between the VFMW and the VDH. It can be read and written by both the VDH and VFMW.

The VDH and VFMW interact with each other in the following two modes:

- The VDH and VFMW interact with each other to decode streams by slices. Specifically, the VFMW decodes the slice header and the streams before the slice header, and the VDH decodes the slice data and the streams after the slice data.
- During multi-channel decoding, the VFMW enables the VDH by frames in time-division multiplexing (TDM) mode.

The video decoding process is as follows:

Step 1 Create a video decoder, and initialize it.

Step 2 Store streams into the stream buffer.

Step 3 Obtain images over the image output interface of the VFMW.

Step 4 Release the buffer that stores the images over the frame recycle interface of the VFMW after the images are displayed.

Step 5 Repeat step 2 to step 4 until decoding is complete.

Step 6 Destroy the video decoder after all streams are played.

----End

8.2 JPGD

8.2.1 Overview

The Joint Photographic Experts Group decoder (JPGD) in an HD chip supports JPEG and MJPEG picture decoding.

8.2.2 Features

The JPGD has the following features:

- Advanced eXtensible interface (AXI) and advanced peripheral bus (APB) interface
- Interrupts
- ITU-T81 baseline profile decoding. That is:
 - Decoding in five YUV JPEG picture formats: YUV 4:0:0, YUV4:2:0, YUV4:2:2 1x2, YUV4:2:2 2x1, and YUV 4:4:4
 - A maximum of four Huffman tables including two direct coefficient (DC) tables and two alternating coefficient (AC) tables
 - A maximum of three quantization tables
 - Sequential decoding
 - Discrete cosine transform-based (DCT-based) JPEG decoding
 - 8-bit depth sampling



- Interleaved scanning
- 1/2, 1/4, or 1/8 frequency domain scaling, significantly reducing memory and bandwidth usage during decoding
- Decoding of static pictures with the resolution of 8096 x 8096 at the maximum and 1 x 1 at the minimum
- Semi-planar output storage format with the maximum resolution of 8096 x 8096
- Decoding of compressed streams by segment
- 1-channel Motion-JPEG 1080p@40 fps, satisfying various decoding systems that have high requirements on real-time performance
- AC or DC decoding error reporting
- Conversion from YUV to RGB
 - Allows the 4:0:0, 4:2:0, 4:2:2 2x1, and 4:2:2 1x2 formats to be up-sampled as the 4:4:4 format.
 - Supports the maximum resolution of 8096 x 8096 and minimum resolution of 8 x 8.
 - Supports the ARGB8888 and ABGR8888 output formats.
 - Supports configurable global alpha.
 - Allows the output to be cropped in any position and in any size.
 - Allows the output to be written to any address of the DDR (4-byte-aligned).
- Standby processing
 - YUV output standby
 - RGB output standby

8.3 PGD

8.3.1 Overview

The PNG and GIF decoder (PGD) is a hardware acceleration module that works with the software to decode PNG and GIF pictures.



Currently, PNG pictures are decoded by hardware, and GIF pictures are decoded by software.

8.3.2 Features

The PGD has the following features:

- APB and AXI interfaces
 - Provides an APB interface for configuring registers.
 - Provides an AXI for reading or writing data.
 - The internal working clock and AXI clock are the same.
- Picture size
 - Supports the maximum resolution of 8191 x 8191.
 - Supports the minimum resolution of 5 x 5 for decoding of interlaced pictures.
 - Supports the minimum resolution of 1 x 1 for decoding of non-interlaced pictures.
- Chunk processing



- Decodes the image data (IDAT) chunks.
- Discards the chunks excluding IDAT chunks.
- Performs CRC check on IDAT chunks.
- Decompression of Z-lib streams (including Huffman decoding and deflate decoding)
 - Supports the Z-lib stream input format.
 - Decompresses the data that is compressed in three Huffman compression modes including dynamic Huffman compression mode, fixed Huffman compression mode, and non-compression mode.
 - Supports the deflate decoding by using the maximum 32 KB deflate window.
 - Performs Adler32 check on the decompressed Z-lib streams (the data in IDAT chunks forms a Z-lib stream).
 - Does not support the dictionary (dictid).
- Pass convergence
 - Pass convergence is not supported by the hardware.
- Line filtering
 - Supports five filtering modes: non-filtering mode, upper adjacent filtering mode, left adjacent filtering mode, averaging filtering mode (referencing the values of upper adjacent and left adjacent points), and adaptive filtering mode.
 - Filters the lines of interlaced or non-interlaced pictures.
 - Filters PNG pictures in all color formats.
- Data format conversion
 - Supports the following input formats: Gray: 1 bit/2 bits/4 bits/8 bits/16 bits, BRG: 8 bits/16 bits, CLUT: 1 bit/2 bits/4 bits/8 bits, Agray: 8 bits/16 bits, and ABGR: 8 bits/16 bits.
 - Reverts the pixel sequence of one byte in the format of Gray 1 bit/2 bits/4 bits or CLUT 1 bit/2 bits/4 bits.
 - Converts the format of Gray 1 bit/2 bits/4 bits into Gray 8 bits by stuffing with 0 or gray value.
 - Converts the format of Gray 8 bits/16 bits into BGR 8 bits/16 bits, and converts the format of AGray 8 bits/16 bits into ABGR 8 bits/16 bits.
 - Converts the format of Gray 8 bits/16 bits into AGray 8 bits/16 bits, converts the format of BGR 8 bits/16 bits into ABGR 8 bits/16 bits, and sets the transparent color.
 - Converts the format of AGray 8 bits/16 bits into Gray 8 bits/16 bits, and converts the format of ABGR 8 bits/16 bits into BGR 8 bits/16 bits.
 - Converts the format of BGR 8 bits/16 bits into RGB 8 bits/16 bits, and converts the format of ABGR 8 bits/16 bits into ARGB 8 bits/16 bits.
 - Converts the format of AGray 8 bits/16 bits into GrayA 8 bits/16 bits, and converts the format of ABGR/ARGB 8 bits/16 bits into BGRA/RGBA 8 bits/16 bits.
 - Converts a 16-bit component into a 8-bit component by discarding lower eight bits or based on the format H8 + ((L8 - H8 > 128) ? 1:0).
 - Converts the format of AGRB8888 into ARGB4444, ARGB1555, RGB 565, RGB 555, and RGB 444 by discarding lower bits.
 - The final effect is displayed after the preceding effects are added in sequence. The effect of each filter can be bypassed. If the input format does not match a filter, its filtering effect is bypassed automatically.
- PNG premultiply



- Data extraction after the decompression of Z-lib streams

The PGD needs to support Z-lib decoding for kernel decoding. The hardware does not support the stream resuming function. If the specified stream length is exceeded during decoding, the hardware stops writing data and reports an error.
- Error tolerance
 - Reports the CRC error.
 - Reports the Adler32 error.
 - Supports the Z-lib stream, and reports the error of the cm field, cinfo field, fdict field, btype field, lenth field, hlit field, or fdict field.
 - Reports the error that occurs during Huffman decoding.
 - Dynamically reports the error that occurs during Huffman copy.
 - Reports the filter type error.
 - Supports the rounding control (RDC) module, and reports the buffer overflow error.
 - Supports the filter (FLT) module, and reports the buffer overflow error.
 - Supports the format (FMT) module, and reports the buffer overflow error.
- Software and hardware interfaces
 - Sets the information about a picture including the picture width, picture height, bit depth, color format, interlaced mode, filtering mode, and compression mode.
 - Sets the start and end addresses for a stream buffer.
 - Sets the start and end addresses for storing streams.
 - Sets the start address, line width, and end address (Z-lib decoding) for storing the target data.
 - Sets the address and line width for storing pass data.
 - Sets the start address for the deflate window buffer.
 - Sets the start and end addresses for filtering upper adjacent data.
 - Selects the PNG decoding or Z-lib decoding mode.
 - Starts PNG decoding.
 - Continues PNG decoding after streams are resumed.
 - Supports the resuming stream interrupt, decoding completion interrupt, and decoding error interrupt.
 - Masks the resuming stream interrupt, decoding completion interrupt, and decoding error interrupt.
 - Determines whether to continue decoding when an error occurs.
 - Collects statistics on error chunks.



Contents

9 Graphics Processing	9-1
9.1 TDE.....	9-1
9.1.1 Overview.....	9-1
9.1.2 Function Description.....	9-1
9.2 GPU (3D Acceleration)	9-2
9.3 VPSS	9-2
9.3.1 Overview.....	9-2
9.3.2 Features.....	9-3



9 Graphics Processing

9.1 TDE

9.1.1 Overview

The TDE draws graphics using hardware, which reduces the CPU usage and improves the memory bandwidth utilization. The TDE reads and writes bitmap data, filtering and scaling coefficients, parameters of linked list nodes, and linked lists over the advanced eXtensible interface (AXI) master bus, and obtains register configuration information over the advanced peripheral bus (APB) slave interface.

The TDE processes graphics using Source 1 and Source 2 as follows:

- Source 1 implements direct copying and direct filling functions during single-source operations.
- Source 2 implements complex functions, such as scaling and anti-flicker, during single-source operations.
- Source 1 and Source 2 collaborate to implement operations such as color blending and raster operation (ROP), and to process images in the macroblock format.
- Source 1 bitmap and Output bitmap supports images in the following formats: 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, and YCbCr444MB.
- Source 2 bitmap supports images in the following formats: RGB444, RGB555, RGB565, RGB888, ARGB4444, ARGB1555, ARGB8565, ARGB8888, CLUT1, CLUT2, CLUT4, CLUT8, ACLUT44, ACLUT88, A1, A8, YCbCr888, AYCbCr8888, YCbCr422, YCbCr400MB, YCbCr422MBH, YCbCr422MBV, YCbCr420MB, and YCbCr444MB.

9.1.2 Function Description

The TDE has the following features:

- Only the little endian system
- Configurable formats for Source 1 bitmap, Source 2 bitmap, and Output bitmap
- Gamma correction and contrast and luminance adjustment
- Color lookup tables (CLUTs)
- Conversion between RGB and YCbCr



- Direct copying
- Direct filling
- 2D resize
- Anti-flicker
- Clipping
- Alpha blending
- ROP
- Colorkey operation
- Programmable scanning mode
- Clip mask
- Software interfaces in linked list mode
- Status interrupts

9.2 GPU (3D Acceleration)

The GPU provides the OpenGL ES2.0/1.1 and OpenVG1.1 standard graphics APIs and integrates the dedicated 3D graphics acceleration engine to implement functions such as geometry processing, pixel processing, memory management, and power management. It has the following superior graphics processing capabilities:

- Comprehensive programmable geometry and pixel processing algorithms, which rapidly process primitives, textures, pixels, and templates
- Support for various input/output formats (including YCbCr) and special effect processing of decoded video data
- 16xFSAA anti-aliasing
- Dynamic frequency scaling (DFS)

9.3 VPSS

9.3.1 Overview

The video processing subsystem (VPSS) processes videos before encoding or after decoding, including VC1, deinterlacing, deringing, scaling, sharpening, aspect ratio conversion, and rotation.

It has the following features:

- Processing of maximum 4096 x 2304 video source in a single frame
- Maximum two video outputs
- Configuration of register linked lists
- Span of 4 KB boundary
- Input data format of semi-planar 420/422, planar 400/420/422/444/411/410, or package 422 YUYV/YVYU/UYVY
- Output data format of semi-planar 420/422
- Input data format of TILE420/400



- Outstanding configuration
- Memory low-power mode by using clock gating
- Soft reset at any time

9.3.2 Features

- VC1: This function is used to map video data based on specific protocols.
- Deinterlacing: The deinterlacing (DEI) module restores the interlaced video source to the progressive video source.
- Deringing: The deringing (DR) module compensates the high-frequency component loss due to video compression and eliminates the ring on the image edge.
- Scaling: The zoom engine (ZME) module supports one to eight times magnification or minification of images with any resolution.
- Image sharpening: The luma transient improvement (LTI) and chroma transient improvement (CTI) modules extract high-frequency components from images before scaling and compensate frequencies for the images processed by the scaler. In this way, image edges are sharpened and contours are clear.
- Aspect ratio conversion: This function is used to implement the conversion between the 4:3 and 16:9 aspect ratios by adding or deleting picture borders.
- Rotation: Rotation by 90 or 270 degrees is supported. The output data format is semi-planar 420, and the input data format must be semi-planar 420 or semi-planar 422.



Contents

10 Audio/Video Interfaces.....	10-1
10.1 VDP.....	10-1
10.1.1 Function Description.....	10-1
10.2 HDMI TX.....	10-4
10.2.1 Overview.....	10-4
10.2.2 Function Description.....	10-4
10.2.3 Operating Modes.....	10-7
10.3 AIAO.....	10-7
10.3.1 Overview.....	10-7
10.3.2 Function Description.....	10-8



Figures

Figure 10-1 Logic block diagram of the HDMI TX interface	10-5
Figure 10-2 Typical application of the HDMI	10-5
Figure 10-3 Logic block diagram of AIAO	10-9
Figure 10-4 Typical application of the AIAO module	10-10



10 Audio/Video Interfaces

10.1 VDP

10.1.1 Function Description

The main display interface of the video display processor (VDP) has the following features:

NOTE

The VDP for Hi3796M V100 does not contain the G2, G3 graphics surfaces. It only supports the HDMI and CVBS output.

- Maximum 3840 x 2160 output resolution for the main video surface
- Overlapping of four surfaces, including two graphics surfaces (G0 and G1) and two video surfaces (V0 and V1)
- Video processing after overlapping (VP)
- Graphics processing after overlapping (GP0)
- VDC_V0 surface
 - Input data formats: semi-planar YCbCr 4:2:0 and semi-planar YcbCr 4:2:2
 - Variable input/output resolution. The minimum input/output resolution is 32 x 32, the maximum input resolution is 1920 x 2160, and the maximum output resolution is 3840 x 2160.
 - Static frames (mute) and chrominance up sampling
 - Global alpha value
 - CSC between YCbCr and RGB at the video layer and the adjustment of luminance, contrast, hue, and saturation
 - Horizontal/Vertical scaling
 - 8-order horizontal luminance filtering and 4-order horizontal chrominance filtering. For each type of filtering, 32 groups of configurable filtering coefficients are provided.
 - 4-order vertical luminance filtering and 4-order vertical chrominance filtering. For each type of filtering, 32 groups of configurable filtering coefficients are provided.
 - Super resolution (SR)
 - 3D videos (side-by-side, top-and-bottom, and frame packing)
 - LetterBox
- VDC_V1 surface



- Input data formats: semi-planar YCbCr 4:2:0 and semi-planar YCbCr 4:2:2
- Variable input/output resolution, 32 x 32 at the minimum and 1920 x 1080 at the maximum
- 16 regions
- Static frames (mute) and chrominance up sampling
- Global alpha value
- CSC between YCbCr and RGB at the video layer and the adjustment of luminance, contrast, hue, and saturation
- 4-order vertical chrominance up sampling
- 3D videos (side-by-side, top-and-bottom, and frame packing)
- LetterBox

NOTE

The G1 and G0 graphics surfaces of the VDP for Hi3796M V100 support the CLUT format, but G0 and G1 cannot use the CLUT format at the same time.

- **GDC_G0**
 - Various package RGB and YUV data formats. For details, see the user guide of GDC_G0.
 - CLUT data format
 - Variable input/output resolution, 32 x 32 at the minimum and 1920 x 1080 at the maximum
 - Three data extension modes
 - Colorkey processing
 - Premultiplied data formats
 - Global alpha and pixel alpha
 - 3D graphics (side-by-side, top-and-bottom, and frame packing)
- **GDC_G1**
 - Various package RGB and YUV data formats. For details, see the user guide of GDC_G1.
 - Variable input/output resolution, 32 x 32 at the minimum and 1920 x 1080 at the maximum
 - Three data extension modes
 - Colorkey processing
 - Premultiplied data formats
 - Global alpha and pixel alpha
 - 3D graphics (side-by-side, top-and-bottom, and frame packing)
- **GDC_GP0**
 - Variable input/output resolution. The minimum input/output resolution is 32 x 32, the maximum input resolution is 1920 x 1080, and the maximum output resolution is 3840 x 2160.
 - 8-order 8-phase horizontal scaling and 4-order 16-phase vertical scaling
 - Sharpening
 - CSC and adjustment of luminance, contrast, hue, and saturation
 - 3D graphics (side-by-side, top-and-bottom, and frame packing)

The VDP auxiliary display interface has the following features:



NOTE

The VDP auxiliary display interface for Hi3796M V100 supports SD outputs.

- Compliance with the CGMS-A standard, including 480i@60 Hz and 576i@50 Hz
- CVBS output
- (M) NTSC or NTSC-J output
- (B, D, G, H, I) PAL, (N) PAL, (Nc) PAL, or (M) PAL output
- Support for séquentiel couleur à mémoire (SECAM)
- Compliance with the Macrovision7.1 L1 standard
- Support for various vertical blanking interval (VBI) standards: Teletext, Closed Caption, CGMS, WSS, and VPS
- Overlapping of three surfaces, including one graphics surface (G4) and two video surfaces (V3 and V4)
- Processing after graphics overlapping (GP1)

NOTE

The V3/V4/G4 surface of the Hi3796M V100 VDP supports the SD resolution.

- VDC_V3 surface
 - Input data formats: semi-planar YCbCr 4:2:0 and semi-planar YCbCr 4:2:2
 - Variable input/output resolution, 32 x 32 at the minimum and 720 x 576 at the maximum
 - Static frames (mute) and chrominance up sampling
 - Global alpha value
 - CSC between YCbCr and RGB at the video layer and the adjustment of luminance, contrast, hue, and saturation
 - Horizontal/Vertical scaling
 - 8-order horizontal luminance filtering and 4-order horizontal chrominance filtering. For each type of filtering, 32 groups of configurable filtering coefficients are provided.
 - 4-order vertical luminance filtering and 4-order vertical chrominance filtering. For each type of filtering, 32 groups of configurable filtering coefficients are provided.
 - LetterBox
- VDC_V4 surface
 - Input data formats: semi-planar YCbCr 4:2:0 and semi-planar YCbCr 4:2:2
 - Variable input/output resolution, 32 x 32 at the minimum and 720 x 576 at the maximum
 - 16 regions
 - Static frames (mute) and chrominance up sampling
 - Global alpha value
 - CSC between YCbCr and RGB at the video layer and the adjustment of luminance, contrast, hue, and saturation
 - 4-order vertical chrominance up sampling
 - LetterBox
- GDC_G4
 - Various package RGB and YUV data formats. For details, see the user guide of GDC_G4.



- Variable input/output resolution, 32 x 32 at the minimum and 720 x 576 at the maximum
- Three data extension modes
- Colorkey processing
- Premultiplied data formats
- Global alpha and pixel alpha
- 3D graphics (side-by-side, top-and-bottom, and frame packing)
- GDC_GP1
 - Variable input/output resolution, 32 x 32 at the minimum and 720 x 576 at the maximum
 - 8-order 8-phase horizontal scaling and 4-order 16-phase vertical scaling
 - CSC and adjustment of luminance, contrast, hue, and saturation

10.2 HDMI TX

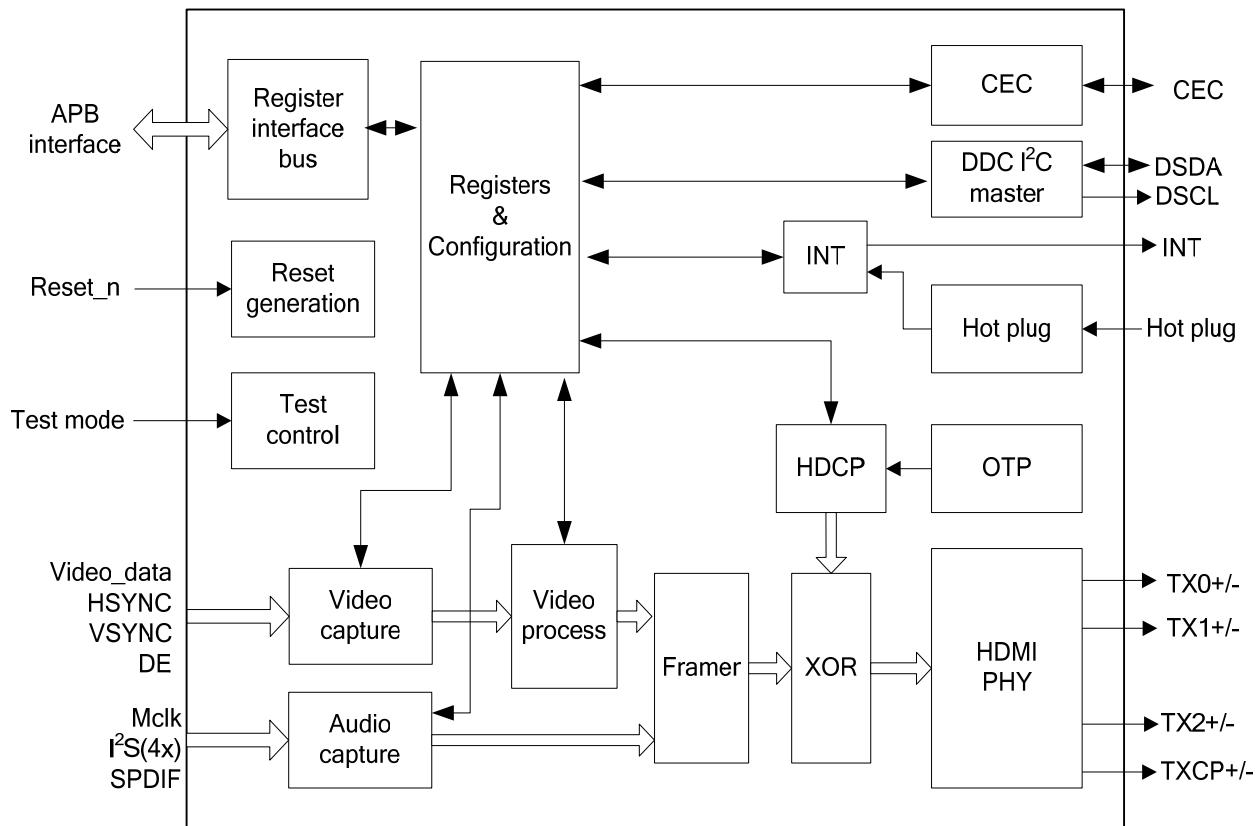
10.2.1 Overview

The HDMI TX interface supports the HDMI 1.4 protocol. It transfers digital audio/video data in a simple, low-cost, but high-performance mode, bringing customers an omni-directional digital experience.

10.2.2 Function Description

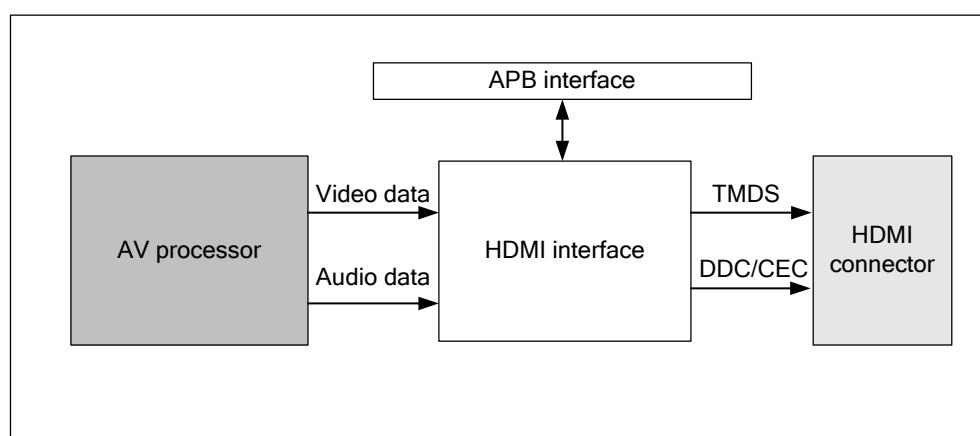
Logic Block Diagram

[Figure 10-1](#) shows the logic block diagram of the HDMI TX interface.

Figure 10-1 Logic block diagram of the HDMI TX interface

Typical Application

Audio/Video data is output to the HDMI connector by configuring corresponding registers over the APB interface based on the input audio/video format of the HDMI interface. [Figure 10-2](#) shows the typical application of the HDMI.

Figure 10-2 Typical application of the HDMI



Features

The HDMI interface has the following features:

- HDMI 1.4a, HDCP 1.4, and DVI 1.0
- DTV video formats from 480i to 4K x 2K
 - 4Kx2K_30 Hz
 - 1080p_60 Hz
 - 1080p_50 Hz
 - 1080p_30 Hz
 - 1080p_25 Hz
 - 1080p_24 Hz
 - 1080i_60 Hz
 - 1080i_50 Hz
 - 720p_60 Hz
 - 720p_50 Hz
 - 576p_50 Hz
 - 480p_60 Hz
 - 576i_50 Hz
 - 480i_60 Hz
- PC video formats from VGA to UXGA
 - 640 x 480_60 Hz
 - 800 x 600_60 Hz
 - 1024 x 768_60 Hz
 - 1280 x 720_60 Hz
 - 1280 x 800_60 Hz
 - 1280 x 1024_60 Hz
 - 1360 x 768_60 Hz
 - 1366 x 768_60 Hz
 - 1400 x 1050_60 Hz
 - 1440 x 900_60 Hz
 - 1600 x 900_60 Hz
 - 1680 x 1050_60 Hz
 - 1920 x 1080_60 Hz
 - 1920 x 1200_60 Hz
 - 2560 x 1600_60 Hz
- 24-/30-bit RGB or YCbCr 4:4:4 data format
- 16-/24-bit YCbCr 4:2:2 data format
- Four I²S interfaces to transfer audio data from a maximum of eight channels
- One SPDIF interface to allow maximum 192 kHz audio transmission
- HDCP encryption for transferring protected audio/video data
- I²C master interface for DDC
- HDMI 3D functions



- 1080p@60 Hz side-by-side (half) mode
- 1080p@60 Hz top-and-bottom mode
- 1080p@60 Hz frame packing mode

10.2.3 Operating Modes

Clock Configuration

If the HDMI TX interface is not used, you can disable its clocks to reduce power consumption.

- To disable all HDMI TX clocks, write 0 to PERI_CRG67 bit[5:0].
- To enable all HDMI TX clocks, write 1 to PERI_CRG67 bit[5:0].

The HDMI TX clocks are enabled by default.

Reset Deassertion

To reset the HDMI TX interface, write 1 to PERI_CRG67 bit[9]. If you want to use the HDMI TX interface, deassert the reset by writing 0 to PERI_CRG67 bit[9]. The HDMI interface is not reset during power-on.

Low-Power Mode

When the HDMI TX interface is not used, you can set it to low-power mode to reduce power consumption. The process is as follows:

- Step 1** Write 0 to RG_TX_EN bit[0] (0xf8ce1804) to disable the PHY output and enable the PHY to enter the low-power mode.
- Step 2** Write 0 to DPD bit[2] (0xf8ce04f4) to enable the HDMI TX controller to enter the low-power mode.

----End

When you want to use the HDMI TX interface, wake it up from the low-power mode by performing the following steps:

- Step 1** Write 1 to DPD bit[2] (0xf8ce04f4) to enable the HDMI TX controller to enter normal mode.
- Step 2** Write 1 to RG_TX_EN bit[0] (0xf8ce1804) to enable the PHY to enter normal mode.

----End

10.3 AIAO

10.3.1 Overview

The audio input and audio output (AIAO) module stores data into the memory based on the required interface timings (I²S/PCM), or transmits audio data to external components over the I²S/PCM/SPDIF interface.



10.3.2 Function Description

Basic Features

The AI module has the following features:

- 8 kHz to 192 kHz sampling rate
- 8-/16-/24-bit sampling precision
- I²S/PCM input
- Configurable PCM data time slot
- 4-wire I²S parallel inputs
- Track mode

The AO module has the following features:

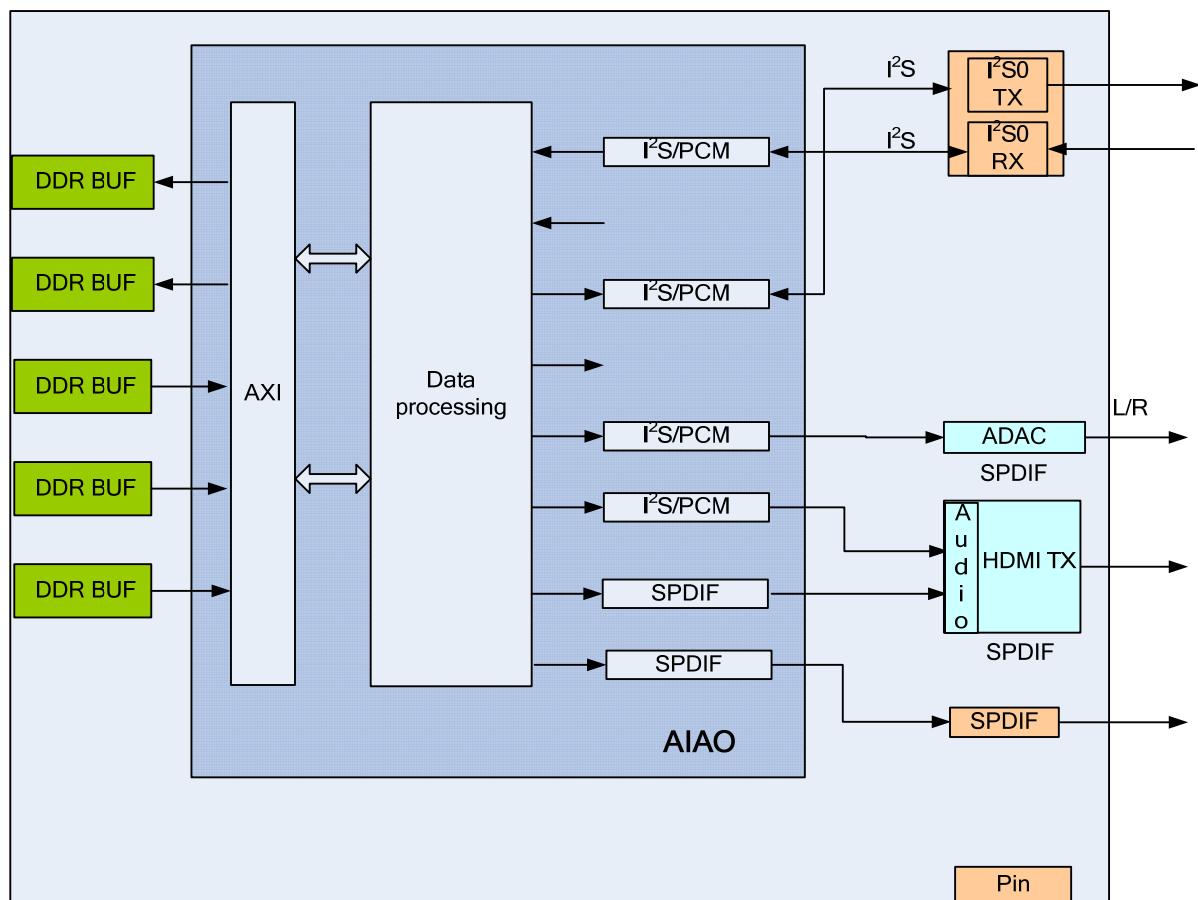
- 8 kHz to 192 kHz sampling rate
- 8-/16-/24-bit sampling precision
- I²S/PCM output
- Configurable PCM data time slot
- 4-wire I²S outputs
- SPDIF digital output
- Track mode
- 4-wire I²S 7.1 audio channel outputs
- Support for operations including mute, unmute, pause, and resume
- Transparent transmission

Logic Block Diagram

[Figure 10-3](#) shows the logic block diagram of the AIAO module.

The AI module receives audio signals, processes the received signals in I²S or PCM mode, and then stores the signals to the DDR. The AO module processes data read from the DDR (processes the volume, fade-out, and track mode) and outputs the data in the I²S, PCM, or SPDIF format.

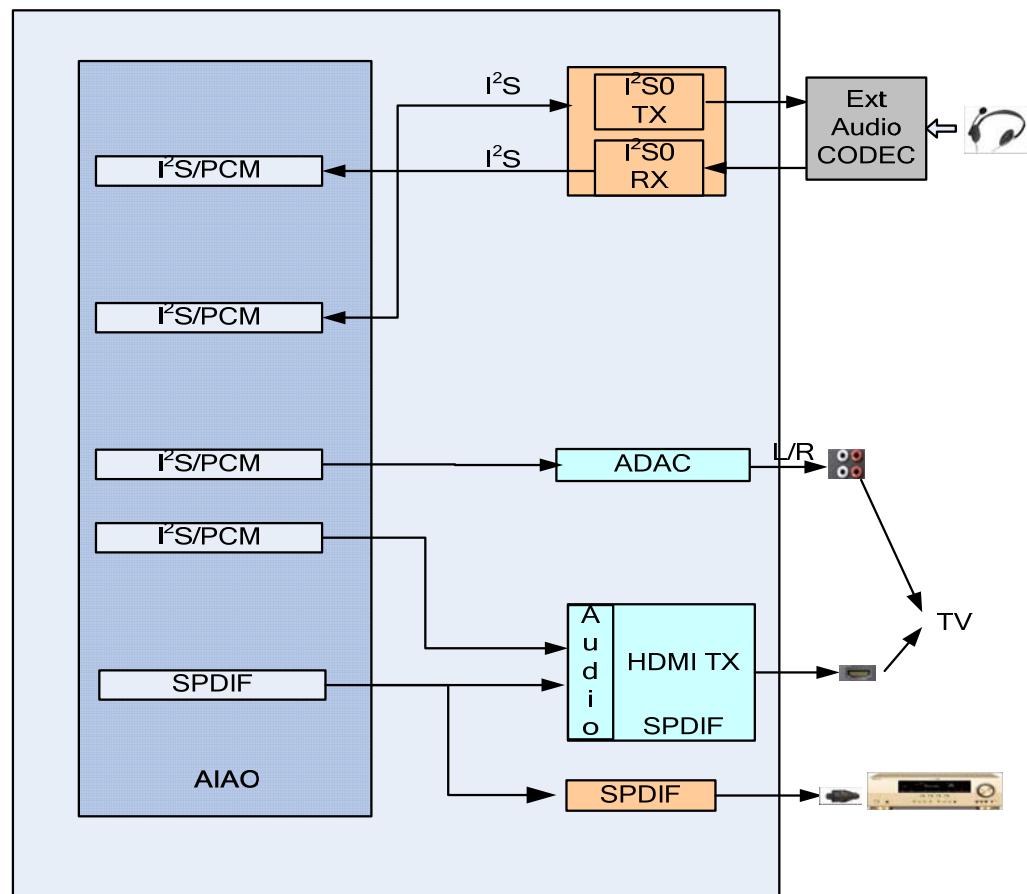
Figure 10-3 Logic block diagram of AIAO



Typical Application

Figure 10-4 shows the typical application of the AIAO module.

Figure 10-4 Typical application of the AIAO module





Contents

11 Peripherals.....	11-1
11.1 GPIO	11-1
11.1.1 Overview	11-1
11.1.2 Features	11-1
11.1.3 Function Description.....	11-2
11.1.4 Operating Mode	11-2
11.1.5 Register Summary.....	11-3
11.1.6 Register Description.....	11-4
11.2 UART	11-9
11.2.1 Overview	11-9
11.2.2 Features	11-9
11.2.3 Function Description.....	11-10
11.2.4 Operating Mode	11-11
11.2.5 Register Summary.....	11-13
11.2.6 Register Description.....	11-14
11.3 I ² C	11-26
11.3.1 Overview	11-26
11.3.2 Function Description.....	11-26
11.3.3 Operating Mode	11-26
11.3.4 Register Summary.....	11-29
11.3.5 Register Description.....	11-29
11.4 IR.....	11-37
11.4.1 Overview	11-37
11.4.2 Features	11-37
11.4.3 Function Description.....	11-37
11.4.4 Operating Mode	11-45
11.4.5 Register Summary.....	11-48
11.4.6 Register Description.....	11-49
11.5 LED	11-64
11.5.1 Overview	11-64
11.5.2 Function Description.....	11-64
11.5.3 Operating Mode	11-65



11.5.4 Interface Signals.....	11-66
11.5.5 Register Summary.....	11-66
11.5.6 Register Description.....	11-67
11.6 SPI.....	11-76
11.6.1 Overview.....	11-76
11.6.2 Features	11-76
11.6.3 Function Description.....	11-76
11.6.4 Peripheral Bus Timings	11-77
11.6.5 Operating Mode	11-84
11.6.6 Register Summary.....	11-86
11.6.7 Register Description.....	11-87
11.7 USB 2.0	11-94
11.7.1 Overview.....	11-94
11.7.2 Function Description.....	11-95
11.7.3 Operating Mode	11-97
11.7.4 Summary of USB Host Register	11-99
11.7.5 Description of USB Host Registers.....	11-99
11.7.6 Offset Address Variables for USB Device Registers.....	11-105
11.7.7 Summary of USB Device Registers.....	11-105
11.7.8 Description of USB Device Registers.....	11-107
11.8 USB 3.0 Host.....	11-178
11.8.1 Overview	11-178
11.8.2 Function Description.....	11-179
11.8.3 Operating Mode	11-181
11.8.4 Register Variables.....	11-181
11.8.5 Register Summary.....	11-182
11.8.6 Register Description.....	11-183
11.9 SCI.....	11-216
11.9.1 Overview	11-216
11.9.2 Features	11-216
11.9.3 Function Description.....	11-216
11.9.4 Operating Mode	11-220
11.9.5 Register Summary.....	11-221
11.9.6 Register Description.....	11-222



Figures

Figure 11-1 Typical application block diagram of the UART	11-10
Figure 11-2 UART frame format	11-11
Figure 11-3 Process for transmitting data by using the I ² C master.....	11-27
Figure 11-4 Process for receiving data by using the I ² C master	11-28
Figure 11-5 Functional block diagram of the IR module.....	11-37
Figure 11-6 Format for transmitting a single frame in NEC with simple repeat code format	11-40
Figure 11-7 Format for transmitting consecutive frames in NEC with simple repeat code format by holding a key down.....	11-40
Figure 11-8 Definitions of bit 0 and bit 1 of the NEC with simple repeat code	11-40
Figure 11-9 Format for transmitting an NEC with simple repeat code.....	11-41
Figure 11-10 Format for transmitting consecutive NEC with simple repeat codes	11-41
Figure 11-11 Format for transmitting a single frame in NEC with full repeat code format.....	11-41
Figure 11-12 Format for transmitting consecutive frames in NEC with full repeat code format by holding a key down.....	11-42
Figure 11-13 Definitions of bit 0 and bit 1 of the NEC with full repeat code	11-42
Figure 11-14 Format for transmitting a single NEC with full repeat code	11-42
Figure 11-15 Format for transmitting a single frame in TC9012 code format.....	11-43
Figure 11-16 Format for transmitting consecutive frames in TC9012 code format by holding a key down ..	11-43
Figure 11-17 Definitions of bit 0 and bit 1 of the TC9012 code.....	11-43
Figure 11-18 Format for transmitting a single TC9012 code	11-43
Figure 11-19 Format for transmitting consecutive TC9012 codes when C0 is 1.....	11-44
Figure 11-20 Format for transmitting consecutive TC9012 codes when C0 is 0.....	11-44
Figure 11-21 Format for transmitting a single frame in SONY code format.....	11-44
Figure 11-22 Format for transmitting consecutive frames in SONY code format by holding a key down	11-45
Figure 11-23 Definitions of bit 0 and bit 1 of the SONY code	11-45
Figure 11-24 Process for initializing the IR module.....	11-46
Figure 11-25 Process for reading the decoded data	11-47



Figure 11-26 Output timing of the LED module	11-65
Figure 11-27 SPI connected to a single slave	11-77
Figure 11-28 Format of a single SPI frame (SPO = 0, SPH = 0).....	11-77
Figure 11-29 Format of consecutive SPI frames (SPO = 0, SPH = 0).....	11-78
Figure 11-30 Format of a single SPI frame (SPO = 0, SPH = 1).....	11-78
Figure 11-31 Format of consecutive SPI frames (SPO = 0, SPH = 1).....	11-79
Figure 11-32 Format of a single SPI frame (SPO = 1, SPH = 0).....	11-79
Figure 11-33 Format of consecutive SPI frames (SPO = 1, SPH = 0).....	11-80
Figure 11-34 SPI single frame format (SPO = 1, SPH = 1).....	11-80
Figure 11-35 SPI consecutive frame format (SPO = 1, SPH = 1)	11-81
Figure 11-36 Format of a single frame for the TI synchronous serial interface	11-81
Figure 11-37 Format of consecutive frames for the TI synchronous serial interface	11-82
Figure 11-38 Format of a single frame for the national semiconductor microwire interface	11-82
Figure 11-39 Format of consecutive frames for the national semiconductor microwire interface	11-83
Figure 11-40 SPI timing	11-84
Figure 11-41 Logic block diagram of the USB 2.0 host controller	11-95
Figure 11-42 Logical block diagram of the USB 2.0 device controller.....	11-96
Figure 11-43 Reference design of the USB 2.0 module	11-96
Figure 11-44 Logic block diagram of the USB 3.0 host.....	11-179
Figure 11-45 USB 3.0 host reference design	11-180
Figure 11-46 Typical application circuit diagram of the SCI.....	11-217
Figure 11-47 Timing for resetting the smart card	11-218
Figure 11-48 Timing for releasing the smart card	11-218
Figure 11-49 SCI timing for soft-resetting the smart card.....	11-219
Figure 11-50 Character transfer structure supported by the SCI	11-219



Tables

Table 11-1 Base addresses for the seven groups of GPIO registers	11-3
Table 11-2 Summary of GPIO registers.....	11-4
Table 11-3 Summary of UART registers	11-13
Table 11-4 Summary of I ² C registers.....	11-29
Table 11-5 Code formats of the received infrared data (NEC with simple repeat code).....	11-38
Table 11-6 Code formats of the received infrared data (NEC with full repeat code).....	11-38
Table 11-7 Code formats of the received infrared data (TC9012 code and SONY code).....	11-39
Table 11-8 Summary of IR registers (base address: 0xF8001000)	11-48
Table 11-9 LED interface signals (8 x 1 matrix keypad scanning)	11-66
Table 11-10 Summary of LED registers (base address: 0xF800_3000).....	11-66
Table 11-11 SPI timing parameters	11-84
Table 11-12 Summary of SPI registers	11-86
Table 11-13 Summary of USB host registers.....	11-99
Table 11-14 Register variables.....	11-105
Table 11-15 Summary of USB device registers (base address: 0xF98C0000).....	11-105
Table 11-16 Variables in the offset addresses for USB 3.0 registers.....	11-181
Table 11-17 Summary of USB 3.0 host registers (base address: 0xF98A_0000)	11-182
Table 11-18 Summary of SCI registers (base address: 0xF8B1_8000)	11-221



11 Peripherals

11.1 GPIO

11.1.1 Overview

Hi3796M V100 supports seven groups of GPIO pins, and each group provides eight programmable input/output pins. Each GPIO pin can be configured as an input or output. These pins are used to generate output signals or capture input signals for specific applications. As an input, each GPIO pin can act as an interrupt source; as an output, each GPIO pin can be separately set to 0 or 1.



CAUTION

For details about the number of GPIO pins, multiplexing relationship between GPIO pins and other pins, and the control modes, see the "Package and Pinout" section in the *Hi3796M V100 Intelligent Network Terminal Media Processor Hardware User Guide*.

11.1.2 Features

The GPIO module has the following features:

- Each GPIO pin can be set to input or output.
 - When a GPIO pin acts as an input pin, it can be used as the interrupt source. Each GPIO pin supports independent interrupt control.
 - When a GPIO pin acts as an output pin, it can be separately cleared or set to 1.
- The GPIO interrupts are controlled by seven registers (such as [GPIO_IS](#)). These registers are used to specify the interrupt source, interrupt edge (falling edge or rising edge), and interrupt trigger modes (level-sensitive mode or edge-sensitive mode). For details about the corresponding interrupt registers of the GPIO, see section 3.6 "Interrupt System".
 - When multiple interrupts are generated at the same time, these interrupts are combined as one interrupt and then reported. For details about the GPIO interrupt mapping, see section 3.6 "Interrupt System".
 - The [GPIO_IS](#), [GPIO_IBE](#), and [GPIO_IEN](#) registers determine the features of the interrupt source and the interrupt trigger type.



[GPIO_RIS](#) and [GPIO_MIS](#) are used to read the raw interrupt status and masked interrupt status respectively. [GPIO_IE](#) controls the final report status of each interrupt. In addition, [GPIO_IC](#) is provided for clearing the interrupt status.

11.1.3 Function Description

Each GPIO pin group contains eight programmable I/O pins. Each GPIO pin can be configured as an input or output. These pins are used to collect input signals or generate output signals for specific purposes.

The GPIO module can generate maskable interrupts based on the level or transition value. The general purpose input output interrupt (GPIOINTR) signal provides an indicator to the interrupt controller, indicating that an interrupt is generated.

11.1.4 Operating Mode

Interface Reset

During POR, all registers are cleared, and therefore the pins work as input pins by default.

When the reset signal is valid, the GPIO status is as follows:

- The interrupt becomes invalid after the corresponding bit of [GPIO_IE](#) is cleared.
- All registers are cleared.
- All pins are configured as inputs.
- All raw interrupt registers are cleared.
- The interrupt trigger mode is configured as edge-sensitive mode.

GPIO

Each pin can be configured as input or output. To configure a GPIO pin, perform the following steps:

Step 1 Configure the corresponding bit of the GPIO pin to enable required GPIO pin functions. For details, see the "Package and Pinout" section in the *Hi3796M V100 Intelligent Network Terminal Media Processor Hardware User Guide*.

Step 2 Set the GPIO pin as input or output by configuring [GPIO_DIR](#).

- If the GPIO pin is used as input, external signals are transmitted through the GPIO pin. In this case, the values of input signals can be obtained by reading [GPIO_DATA](#).
Note: The input signals are also transmitted to the pins that are multiplexed with the GPIO pin.
- If the GPIO pin is used as output, values are written to [GPIO_DATA](#) and then output through the GPIO pin.
Note: If the GPIO interrupt function is enabled, an interrupt is generated when the output signal meets the triggering condition.

----End

Interrupt Operation

To generate an interrupt but not a pseudo interrupt, perform the following steps:



- Step 1** Select the edge-sensitive mode or level-sensitive mode by configuring [GPIO_IS](#).
- Step 2** Select the falling-/rising-edge-sensitive mode or high-/low-level-sensitive mode by configuring [GPIO_IEV](#).
- Step 3** If the edge-sensitive mode is selected, select single-edge-sensitive mode or dual-edge-sensitive mode by configuring [GPIO_IBE](#).
- Ensure that the GPIO data lines are stable during preceding operations.
- Step 4** Write 0xFF to [GPIO_IC](#) to clear the interrupt.
- Step 5** Set [GPIO_IE](#) to 1 to enable the interrupt.
- End

The GPIO interrupts are controlled by seven registers. When one or more GPIO pins generate interrupts, a combined interrupt is output to the interrupt controller. The following are the differences between the edge-sensitive mode and the level-sensitive mode:

- Edge-sensitive mode: Software must clear the interrupt to enable superior interrupts.
- Level-sensitive mode: The external interrupt source should keep the level until the processor identifies the interrupt.

11.1.5 Register Summary

[Table 11-1](#) lists the base addresses for the seven groups of GPIO registers.

Table 11-1 Base addresses for the seven groups of GPIO registers

Register	Base Address
GPIO0	0xF8B2_0000
GPIO1	0xF8B2_1000
GPIO2	0xF8B2_2000
GPIO3	0xF8B2_3000
GPIO4	0xF8B2_4000
GPIO5	0xF800_4000
GPIO6	0xF8B2_6000

The GPIO groups in table [Table 11-1](#) have the same registers.

 **NOTE**

- The address for the register corresponding to GPIO_n consists of the GPIO_n base address and register offset address.
- n ranges from 0 to 6.

[Table 11-2](#) describes GPIO registers.



Table 11-2 Summary of GPIO registers

Offset Address	Register	Description	Page
0x000–0x3FC	GPIO_DATA	GPIO data register	11-4
0x400	GPIO_DIR	GPIO direction control register	11-5
0x404	GPIO_IS	GPIO interrupt trigger register	11-5
0x408	GPIO_IBE	GPIO interrupt edge control register	11-6
0x40C	GPIO_IEV	GPIO interrupt trigger event register	11-6
0x410	GPIO_IE	GPIO interrupt mask register	11-7
0x414	GPIO_RIS	GPIO raw interrupt status register	11-7
0x418	GPIO_MIS	GPIO masked interrupt status register	11-8
0x41C	GPIO_IC	GPIO interrupt clear register	11-8
0x420	GPIO_RESERVED	Reserved GPIO register	11-9

11.1.6 Register Description

GPIO_DATA

GPIO_DATA is a GPIO data register. It is used to buffer the input or output data.

When the corresponding bit of [GPIO_DIR](#) is configured as output, the values written to GPIO_DATA are output to the corresponding pin (ensure that the configuration of pin multiplexing is correct). If the bit is configured as input, the value of the corresponding input pin is read.



CAUTION

When the corresponding bits of [GPIO_DIR](#) are set to inputs, the pin values are returned if the read operation is valid. When the corresponding bits are set to outputs, the written values are returned if the read operation is valid.

The GPIO_DATA register masks the read and write operations on bits by using PADDR[9:2]. This register corresponds to 256 address spaces. PADDR[9:2] correspond to GPIO_DATA[7:0]. When a PADDR bit is high, the corresponding bit in GPIO_DATA can be read or written; when the corresponding bit is low, the read or write operation is not allowed. For example:

- If the address is 0x3FC (0b11_1111_1100), operations on all the eight bits of GPIO_DATA[7:0] are valid.



- If the address is 0x200 (0b10_0000_0000), the operation on only 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	GPIO input data when the GPIO is configured as input or GPIO output data when the GPIO is configured as output Each bit can be separately controlled. This register works with GPIO_DIR.									

GPIO_DIR

GPIO_DIR is a GPIO direction control register. It is used to configure the direction of each 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. Bit[7:0] correspond to GPIO_DATA bit[7:0] respectively. Each bit can be separately controlled. 0: input 1: output									

GPIO_IS

GPIO_IS is a 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. Bit[7:0] correspond to GPIO_DATA bit[7:0] respectively. Each bit can be separately controlled. 0: edge-sensitive mode 1: level-sensitive mode						

GPIO_IBE

GPIO_IBE is a GPIO interrupt edge control register. It is used to configure the edge trigger mode of each GPIO pin.

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. Bit[7:0] correspond to GPIO_DATA[7:0] respectively. Each bit can be separately controlled. 0: single-edge-sensitive mode. GPIO_IEV controls whether the interrupt is rising-edge-sensitive or falling-edge-sensitive. 1: dual-edge-sensitive mode						

GPIO_IEV

GPIO_IEV is a GPIO interrupt trigger event register. It is used to configure the interrupt trigger event of a GPIO pin.



Offset Address								Register Name	Total Reset Value
Bit	0x40C							GPIO_IEV	0x00
Name	gpio_iev								
Reset	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description						
[7:0]	RW	gpio_iev	GPIO interrupt trigger event. Bit[7:0] correspond to GPIO_DATA bit[7:0] respectively. Each bit can be separately controlled. 0: The interrupt is triggered by the falling edge or low level. 1: The interrupt is triggered by the rising edge or high level.						

GPIO_IE

GPIO_IE is a GPIO interrupt mask register. It is used to mask GPIO interrupts.

Offset Address								Register Name	Total Reset Value
Bit	0x410							GPIO_IE	0x00
Name	gpio_ie								
Reset	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description						
[7:0]	RW	gpio_ie	GPIO interrupt mask. Bit[7:0] correspond to GPIO_DATA bit[7:0] respectively. Each bit can be separately controlled. 0: masked 1: not masked						

GPIO_RIS

GPIO_RIS is a GPIO raw interrupt status register. It is used to query the raw interrupt status of each GPIO pin.



Offset Address								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. Bit[7:0] correspond to GPIO_DATA bit[7:0] respectively, indicating the unmasked interrupt status. This status is not under the mask control of the GPIO_IE register. 0: No interrupt is generated. 1: An interrupt is generated.						

GPIO_MIS

GPIO_MIS is a GPIO masked interrupt status register. It is used to query the masked interrupt status of each GPIO pin.

Offset Address								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. Bit[7:0] correspond to GPIO_DATA bit[7:0] respectively, indicating the masked interrupt status. This status can be masked and controlled by the GPIO_IE register. 0: The interrupt is invalid. 1: The interrupt is valid.						

GPIO_IC

GPIO_IC is a GPIO interrupt clear register. It is used to clear the [GPIO_RIS](#) and [GPIO_MIS](#) registers and interrupts generated by GPIO pins.



Offset Address								Register Name	Total Reset Value
Bit	0x41C							GPIO_IC	0x00
Name	gpio_ic								
Reset	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description						
[7:0]	WC	gpio_ic	GPIO interrupt clear. Bit[7:0] correspond to GPIO_DATA bit[7:0] respectively. Each bit can be separately controlled. 0: not cleared 1: cleared						

GPIO_RESERVED

GPIO_RESERVED is a reserved GPIO register. It must be configured as required.

Offset Address								Register Name	Total Reset Value
Bit	0x420							GPIO_RESERVED	0x00
Name	reserved								
Reset	0	0	0	0	0	0	0	0	0
Bits	Access	Name	Description						
[7:0]	RW	reserved	This field must be set to 0x00.						

11.2 UART

11.2.1 Overview

The UART interface is an asynchronous serial communication interface. The UART is mainly used to interconnect with the UART of an external chip so that the two chips can communicate with each other.

Hi3796M V100 provides three UART units. UART 0 and UART 1 support the 2-wire mode, and UART 2 supports the 4-wire mode.

11.2.2 Features

The UART module has the following features:

- Supports a 16x8-bit TX FIFO and a 16x12-bit RX FIFO.



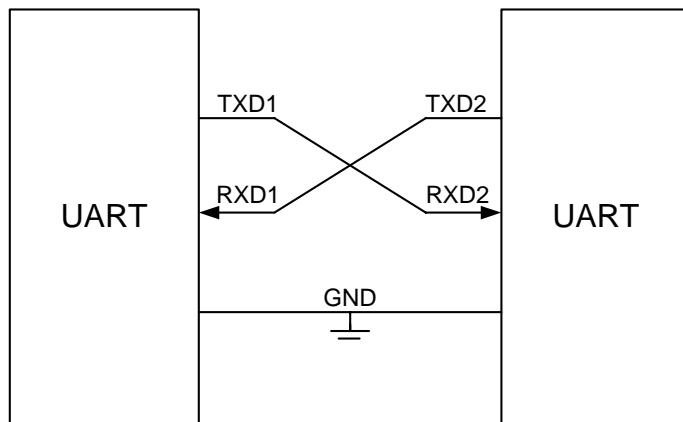
- Allows the widths of the data bit and stop bit to be programmed. The width of the data bit can be set to 5 bits, 6 bits, 7 bits, or 8 bits and that of the stop bit can be set to 1 bit or 2 bits by programming.
- Supports parity check or no check.
- Allows the transfer rate to be programmed.
- Supports RX FIFO interrupts, TX FIFO interrupts, RX timeout interrupts, and error interrupts.
- Supports querying of the status of raw and masked interrupts.
- Allows the UART or the RX/TX function of the UART to be disabled by programming, reducing power consumption.
- Allows the UART clock to be disabled to reduce power consumption.

11.2.3 Function Description

Application Block Diagram

Figure 11-1 shows the typical application block diagram the UART.

Figure 11-1 Typical application block diagram of the UART



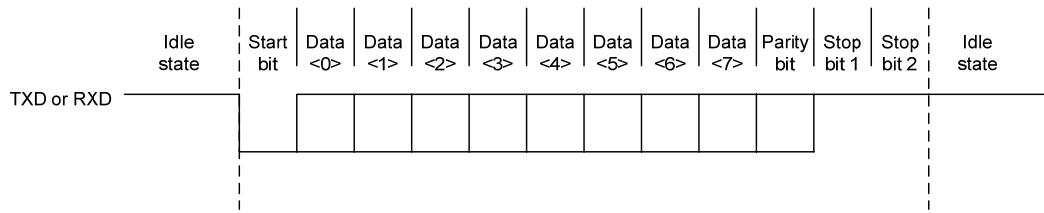
The UART serves as an asynchronous bidirectional serial bus. It provides a simple and effective data transfer mode that requires only two data lines.

Function Implementation

One frame transfer of the UART involves the start signal, data signal, parity bit, and stop signal, as shown in Figure 11-2. The data frame is output from the TxD port of one UART and then is input to the RxD port of the other UART.



Figure 11-2 UART frame format



The following describes the start signal, data signal, parity bit, and stop signal:

- Start signal (start bit)

It is the start flag of a data frame. According to the UART protocol, a low level of the TXD signal indicates the start of a data frame. When the UART does not transmit data, the level must remain high.

- Data signal (data bit)

The data bit width can be set to 5 bits, 6 bits, 7 bits, or 8 bits based on the actual application scenario.

- Parity bit

It is a 1-bit error correction signal. The UART parity bit can be an odd parity bit, even parity bit, or stick parity bit. The parity bit can be enabled or disabled. For details, see the description of [UART_LCR_H](#).

- Stop signal (stop bit)

It is the stop bit of a data frame. The stop bit width can be set to 1 bit or 2 bits. The high level of TXD indicates the end of a data frame.

11.2.4 Operating Mode

11.2.4.1 Configuring the Baud Rate

The operating baud rate of the UART can be set by configuring [UART_IBRD](#) and [UART_FBRD](#). The baud rate is calculated as follows:

$$\text{Current baud rate} = \text{Frequency of the UART reference clock} / (16 \times \text{Clock divider})$$

The clock divider consists of the integer part and the decimal part that correspond to [UART_IBRD](#) and [UART_FBRD](#) respectively.

For example, if the frequency of the internal bus clock is 60 MHz, [UART_IBRD](#) is set to 0x1E, and [UART_FBRD](#) is set to 0x00, the current baud rate is 0.125 Mbit/s ($60/(16 \times 30)$).

The typical UART baud rates are 9600 bit/s, 14,400 bit/s, 19,200 bit/s, 38,400 bit/s, 57,600 bit/s, 76,800 bit/s, 115,200 bit/s, 230,400 bit/s, and 460,800 bit/s.

The following examples illustrate how to calculate the clock divider and how to configure the clock divider register.

If the required baud rate is 230,400 bit/s and the frequency of [UARTCLK](#) is 100 MHz, the clock divider is calculated as follows: Clock divider = $(100 \times 10^6) / (16 \times 230,400) = 27.1267$. Therefore, IBRD (the integer part) is 27, and FBRD (the decimal part) is 0.1267.

Calculate the value of 6-bit [UART_FBRD](#) by using the following formula: m = integer ($FBRD \times 2^n + 0.5$). n indicates the width of [UART_FBRD](#). According to the preceding results,



$m = \text{integer} (0.1267 \times 2^6 + 0.5) = 8$. Then set **UART_IBRD** to 0x001B and **UART_FBRD** to 0x08.

If the decimal part of the clock divider is set to 8, the actual divisor is 27.125 ($27 + 8/64$), the baud rate and error rate are calculated as follows:

- Baud rate = $(100 \times 10^6)/(16 \times 27.125) = 230,414.75$
- Error rate = $(230,414.75 - 230,400)/230,400 \times 100 = 0.006\%$.

The maximum error rate is 1.56% ($1/64 \times 100$) when the 6-bit **UART_FBRD** is used. If m is 1, the error rate is accumulated for more than 64 clock cycles.

11.2.4.2 Soft-Resetting the UART Controller

The UART controller can be independently reset by configuring CRG registers.

- The UART1 controller can be independently soft-reset by setting **SC_CLKGATE_SRST_CTRL[uart_srst_req]** to 1.
- The UART0 controller can be independently soft-reset by setting **PERI_CRG26[uart0_srst_req]** to 1.
- The UART2 controller can be independently soft-reset by setting **PERI_CRG26[uart2_srst_req]** to 1.

After reset, the configuration registers are restored to default values. Therefore, these registers must be initialized and configured again.

11.2.4.3 Transmitting Data in Interrupt or Query Mode

Initializing the UART

The UART is initialized as follows:

- Step 1** Write 0 to **UART_CR** bit[0] to disable the UART.
- Step 2** Write 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 TX and RX FIFO thresholds.
- Step 5** If the driver runs in interrupt mode, set **UART_IMSC** to enable corresponding interrupt signals. If the driver runs in query mode, disable the generation of corresponding interrupts.
- Step 6** Write 1 to **UART_CR** bit[0] to enable the UART.

----End

Transmitting Data

Data is transmitted as follows:

- Step 1** Write the data to be transmitted to **UART_DR** to start data transmission.
- Step 2** In query mode, check the TX_FIFO status by reading **UART_FR[5]** if data is being consecutively transmitted. According to the TX_FIFO status, determine whether to transmit data to TX_FIFO. In interrupt mode, check the TX_FIFO status by reading the corresponding interrupt status bits and then determine whether to transmit data to TX_FIFO.



Step 3 Check whether the UART transmits all data by reading [UART_FR](#) bit[7]. If [UART_FR](#) bit[7] is 1, data transmission is complete.

----End

Receiving Data

Data is received as follows:

- In query mode, check the RX FIFO status by reading [UART_FR](#)[rxfc] during data reception, and then determine whether to read data from the RX FIFO based on its status.
- In interrupt mode, determine whether to read data from the RX FIFO based on corresponding interrupt status bits.

11.2.5 Register Summary

Hi3796M V100 has three UARTs. The following are the base addresses for corresponding UART registers:

- UART0: 0xF8B0_0000
- UART1: 0xF800_6000
- UART2: 0xF8B0_2000

[Table 11-3](#) describes UART registers.

Table 11-3 Summary of UART registers

Offset Address	Register	Description	Page
0x000	UART_DR	UART data register	11-14
0x004	UART_RSR	RX status register or error clear register	11-14
0x018	UART_FR	UART flag register	11-15
0x024	UART_IBRD	Integral baud rate register	11-17
0x028	UART_FBRD	Decimal baud rate register	11-17
0x02C	UART_LCR_H	Transfer mode control register	11-18
0x030	UART_CR	UART control register	11-19
0x034	UART_IFLS	Interrupt FIFO threshold select register	11-20
0x038	UART_IMSC	Interrupt mask register	11-21
0x03C	UART_RIS	Raw interrupt status register	11-22
0x040	UART_MIS	Masked interrupt status register	11-23
0x044	UART_ICR	Interrupt clear register	11-24
0x048	UART_DMACR	DMA control register	11-25



11.2.6 Register Description

UART_DR

UART_DR is a UART data register that stores the received data and data to be transmitted. The RX status can be queried by reading this register.

		Offset Address				Register Name		Total Reset Value									
		0x000				UART_DR		0x0000									
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]	-	reserved			Reserved												
[11]	RO	oe			Overflow error 0: No overflow error occurs. 1: An overflow error occurs. That is, a data segment is received when the RX FIFO is full.												
[10]	RO	be			Break error 0: No break error occurs. 1: A break error occurs. That is, the RX data input signal retains low longer than a full word transfer. A full word consists of a start bit, data bits, a parity bit, and a 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 an RX status register or error clear register.

- It acts as the RX status register when being read.
- It acts as the error clear register when being written.

You can obtain the RX status by reading [UART_DR](#). 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](#) updates faster than that in [UART_RSR](#).



CAUTION

Writing any value to UART_RSR resets 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]	-	reserved	Reserved						
[3]	RW	oe	Overflow error 0: No overflow error occurs. 1: An overflow error occurs. When the FIFO is full, the contents in the FIFO remain valid. No data will be written to the FIFO and overflow occurs in the shift register. In this case, the CPU must read the data immediately to spare the FIFO.						
[2]	RW	be	Break error 0: No break error occurs. 1: A break error occurs. If the RX data input signal retains low longer than a full word transfer, a break error is considered. A full word consists of a start bit, data bits, a parity bit, and a 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 a UART flag register.



Offset Address										Register Name					Total Reset Value						
Bit	0x018										UART_FR					0x0197					
Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
	reserved										txfe	txff	txff	rxfe	busy	reserved					
Reset	0	0	0	0	0	0	0	1	1	0	0	1	0	1	1	1					
Bits	Access	Name	Description																		
[15:8]	-	reserved	Reserved																		
[7]	RO	txfe	This bit is defined by the status of UART_LCR_H[fen] . If UART_LCR_H[fen] is 0, this bit is set to 1 when the TX holding register is empty. If UART_LCR_H[fen] is 1, this bit is set to 1 when the TX FIFO is empty.																		
[6]	RO	txff	This bit is defined by the status of UART_LCR_H[fen] . If UART_LCR_H[fen] is 0, this bit is set to 1 when the RX holding register is full. If UART_LCR_H[fen] is 1, this bit is set to 1 when the RX FIFO is full.																		
[5]	RO	txff	This bit is defined by the status of UART_LCR_H[fen] . If UART_LCR_H[fen] is 0, this bit is set to 1 when the TX holding register is full. If UART_LCR_H[fen] is 1, this bit is set to 1 when the TX FIFO is full.																		
[4]	RO	rxfe	This bit is defined by the status of UART_LCR_H[fen] . If UART_LCR_H[fen] is 0, this bit is set to 1 when the RX holding register is empty. If UART_LCR_H[fen] is 1, this bit is set to 1 when the RX FIFO is empty.																		
[3]	RO	busy	UART busy/idle status 0: The UART is idle or data transmission is complete. 1: The UART is busy transmitting data. If this bit is set to 1, the status is retained until the entire byte (including all stop bits) is transmitted from the shift register. Regardless of whether the UART is enabled, this bit is set to 1 when the TX FIFO is not empty.																		
[2:0]	-	reserved	Reserved																		



UART_IBRD

UART_IBRD is an integer baud rate register.

Offset Address										Register Name							
0x024										UART_IBRD							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	bauddivint																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name			Description												
[15:0]	RW	bauddivint			Clock divider corresponding to the integral part of the baud rate All bits are cleared after reset.												

UART_FBRD

UART_FBRD is a decimal baud rate register.



CAUTION

- The values of UART_IBRD and UART_FBRD are updated only after the current data transmission or reception is complete.
- The minimum clock divider is 1, and the maximum clock divider is 65,535. That is, UART_IBRD cannot be 0, and UART_FBRD is ignored if UART_IBRD is 0. If UART_FBRD is set to 0x1E and UART_IBRD is set to 0x01, the integral part of the clock divider is 30 and the decimal part of the clock divider is 0.015625. Therefore, the clock divider is 30.015625.
- UART baud rate = Internal bus frequency/(16 x Clock divider) = Internal bus frequency/(16 x 30.015625)

Offset Address										Register Name							
0x028										UART_FBRD							
Bit	7	6	5	4	3	2	1	0									
Name	reserved			banddivfrac													
Reset	0	0	0	0	0	0	0	0									
Bits	Access	Name			Description												
[7:6]	-	reserved			Reserved												
[5:0]	RW	banddivfrac			Clock divider corresponding to the decimal part of the baud rate All bits are cleared after reset.												



UART_LCR_H

UART_LCR_H is a transfer mode control register. The registers UART_LCR_H, [UART_IBRD](#), and [UART_FBRD](#) constitute a 30-bit register. If [UART_IBRD](#) and [UART_FBRD](#) are updated, UART_LCR_H must also be updated.

	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]	-	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 during transmission and detection. When bit 1 and bit 7 are set to 1 and bit 2 is set to 0, the parity bit is 1 during transmission and detection. 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		TX/RX FIFO enable 0: disabled 1: enabled													
[3]	RW	stp2		Whether a 2-bit stop bit exists at the end of a 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 RX logic does not check for the 2-bit stop bit during data reception.													
[2]	RW	eps		Parity bit select during data transmission and reception 0: The odd parity bit is generated or checked during data transmission and reception. 1: The even parity bit is generated or checked during data transmission and reception. This bit is invalid when UART_LCR_H[fen] is 0.													



[1]	RW	pen	Parity check enable 0: The parity check is disabled. 1: The parity bit is generated at the TX end and checked at the RX end.
[0]	RW	brk	Break TX 0: invalid 1: After the current data transmission is complete, the UTXD outputs low level continuously. Note: This bit must retain 1 for at least two full frames to ensure that the break command is executed properly. In normal cases, this bit must be set to 0.

UART_CR

UART_CR is a UART control register.

To configure UART_CR, perform the following steps:

- Step 1** Write 0 to UART_CR[uarten] to disable the UART.
- Step 2** Wait until the current data transmission or reception is complete.
- Step 3** Set [UART_LCR_H\[fen\]](#) to 0.
- Step 4** Configure UART_CR.
- Step 5** Write 1 to UART_CR[uarten] 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	1	1	0	0	0	0	0	0	0	0	0
Bits	Access	Name			Description												
[15]	RW	ctsen			CTS hardware flow control enable 0: disabled 1: enabled. Data can be transmitted only when the nUARTCTS signal is valid.												
[14]	RW	rtsen			RTS hardware flow control enable 0: disabled 1: enabled. The data RX request can be sent only when the RX FIFO has free space.												



[13:12]	-	reserved	Reserved
[11]	RW	rts	Request TX This bit is the inversion of the status output signal nUARTRTS of the UART modem. 0: The output signal is retained. 1: When this bit is set to 1, the output signal is 0.
[10]	RW	dtr	Data TX ready This bit is the inversion of the status output signal nUARTDTR of the UART modem. 0: The output signal is retained. 1: When this bit is set to 1, the output signal is 0.
[9]	RW	rxen	UART RX enable 0: disabled 1: enabled If the UART is disabled during data reception, the current data reception is stopped abnormally.
[8]	RW	txen	UART TX enable 0: disabled 1: enabled If the UART is disabled during data transmission, the current data transmission is stopped abnormally.
[7]	RW	lbe	Loopback enable 0: disabled 1: UARTRXD output is looped back to UARTTXD.
[6:1]	-	reserved	Reserved
[0]	RW	uarten	UART enable 0: disabled 1: enabled If the UART is disabled during data reception and transmission, the data transfer is stopped abnormally.

UART_IFLS

UART_IFLS is an interrupt FIFO threshold register. It is used to set the trigger threshold for the FIFO interrupt (UART_TXINTR or UART_RXINTR).



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	0	1	0		
Bits	Access	Name			Description														
[15:6]	-	reserved			Reserved														
[5:3]	RW	rxiflsel			RX interrupt FIFO threshold. An RX interrupt is triggered when any of the following conditions is met: 000: RX FIFO \geq 1/8 full 001: RX FIFO \geq 1/4 full 010: RX FIFO \geq 1/2 full 011: RX FIFO \geq 3/4 full 100: RX FIFO \geq 7/8 full 101–111: reserved														
[2:0]	RW	txiflsel			TX interrupt FIFO threshold. A TX interrupt is triggered when any of the following conditions is met: 000: TX FIFO \leq 1/8 full 001: TX FIFO \leq 1/4 full 011: TX FIFO \leq 3/4 full 010: TX FIFO \leq 1/2 full 100: TX FIFO \leq 7/8 full 101–111: reserved														

UART_IMSC

UART_IMSC is an interrupt mask register.

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										beim	peim	feim	rtim	txim	reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name			Description														
[15:11]	-	reserved			Reserved														



[10]	RW	oeim	Overflow error interrupt mask 0: masked 1: not masked
[9]	RW	beim	Break error interrupt mask 0: masked 1: not masked
[8]	RW	peim	Parity check interrupt mask 0: masked 1: not masked
[7]	RW	feim	Frame error interrupt mask 0: masked 1: not masked
[6]	RW	rtim	RX timeout interrupt mask 0: masked 1: not masked
[5]	RW	txim	TX interrupt mask 0: masked 1: not masked
[4]	RW	rxim	RX interrupt mask 0: masked 1: not masked
[3:0]	-	reserved	Reserved

UART_RIS

UART_RIS is a raw interrupt status register. The contents of this register are not affected by the UART_IMSC register.

	Offset Address		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	rtiris	ttris	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]	-	reserved			Reserved											



[10]	RO	oeris	Raw overflow 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 check 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 RX timeout interrupt status 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	txris	Raw TX interrupt status 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	rxris	Raw RX interrupt status 0: No interrupt is generated. 1: An interrupt is generated.
[3:0]	-	reserved	Reserved

UART_MIS

UART_MIS is a masked interrupt status register. The values of this register are the results obtained after UART_RIS is ANDed with UART_IMSC.

	Offset Address 0x040										Register Name UART_MIS								Total Reset Value 0x0000
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	reserved						bemis	permis	femis	rtrmis	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]	-	reserved			Reserved														



[10]	RO	oemis	Masked overflow 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 check 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 RX timeout interrupt status 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	txmis	Masked TX interrupt status 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	rxmis	Masked RX interrupt status 0: No interrupt is generated. 1: An interrupt is generated.
[3:0]	-	reserved	Reserved

UART_ICR

UART_ICR is an interrupt clear register. Writing 1 clears the corresponding interrupt, and 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	0 0	0 0	0 0	0 0	0 0
Bits	Access	Name			Description								
[15:11]	-	reserved			Reserved								



[10]	WO	oeic	Overflow error interrupt clear 0: invalid 1: cleared
[9]	WO	beic	Break error interrupt clear 0: invalid 1: cleared
[8]	WO	peic	Parity check interrupt clear 0: invalid 1: cleared
[7]	WO	feic	Error interrupt clear 0: invalid 1: cleared
[6]	WO	rtic	RX timeout interrupt clear 0: invalid 1: cleared
[5]	WO	txic	TX interrupt clear 0: invalid 1: cleared
[4]	WO	rxic	RX interrupt clear 0: invalid 1: cleared
[3:0]	-	reserved	Reserved

UART_DMCR

UART_DMCR is a DMA control register. It is used to enable or disable the DMAs of the TX and RX FIFOs.

	Offset Address 0x048															Register Name UART_DMCR			
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	0		
Bits	Access	Name		Description															
[15:3]	-	reserved		Reserved															



[2]	RW	dmaonerr	DMA enable for the RX channel when the UART error interrupt (UARTEINTR) is generated 0: When UARTEINTR is valid, the DMA output request (UARTRXDMASREQ or UARRTXDMABREQ) of the RX channel is valid. 1: When UARTEINTR is valid, the DMA output request (UARTRXDMASREQ or UARRTXDMABREQ) of the RX channel is invalid.
[1]	RW	txdmae	TX FIFO DMA enable 0: disabled 1: enabled
[0]	RW	rxdmae	RX FIFO DMA enable 0: disabled 1: enabled

11.3 I²C

11.3.1 Overview

The I²C module serves as a slave on the advanced peripheral bus (APB) and a master on the I²C bus. It allows the CPU to read/write data from/to slave devices on the I²C bus.

11.3.2 Function Description

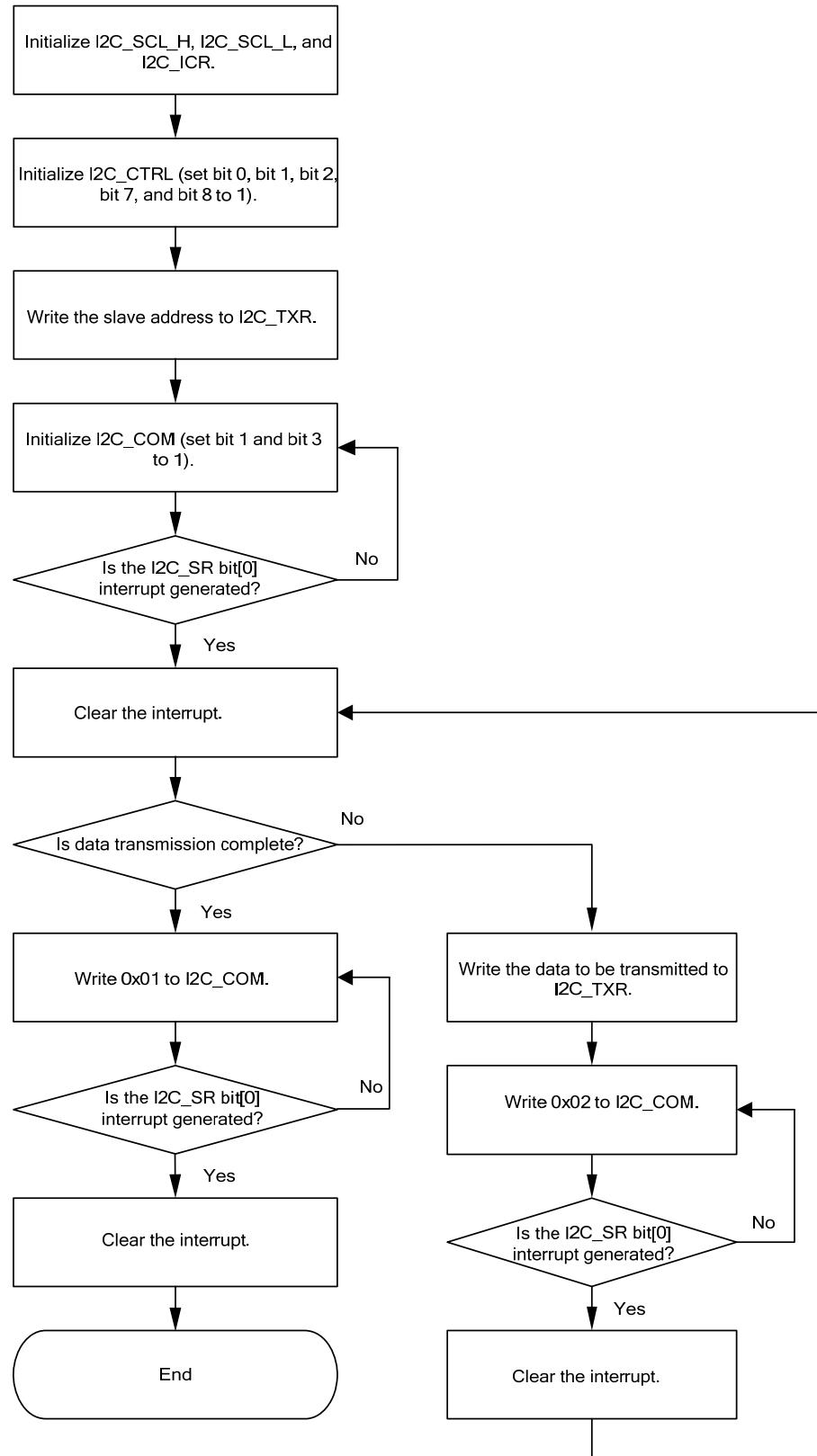
The I²C module has the following features:

- Supports the I²C bus protocol V2.0 and only the master mode.
- Acts as a slave on the APB bus and a master on the I²C bus and supports bus arbitration when there are multiple masters.
- Supports clock synchronization as well as bit and byte waiting.
- Supports the interrupt or polling operation.
- Supports the standard address (7 bits) and extended address (10 bits).
- Works at the standard mode (100 kbit/s) and fast mode (400 kbit/s).
- Supports general call and start byte.
- Does not support the C-Bus device on the I²C bus.
- Filters the received SDA and SCL signals.

11.3.3 Operating Mode

11.3.3.1 Initializing the I²C

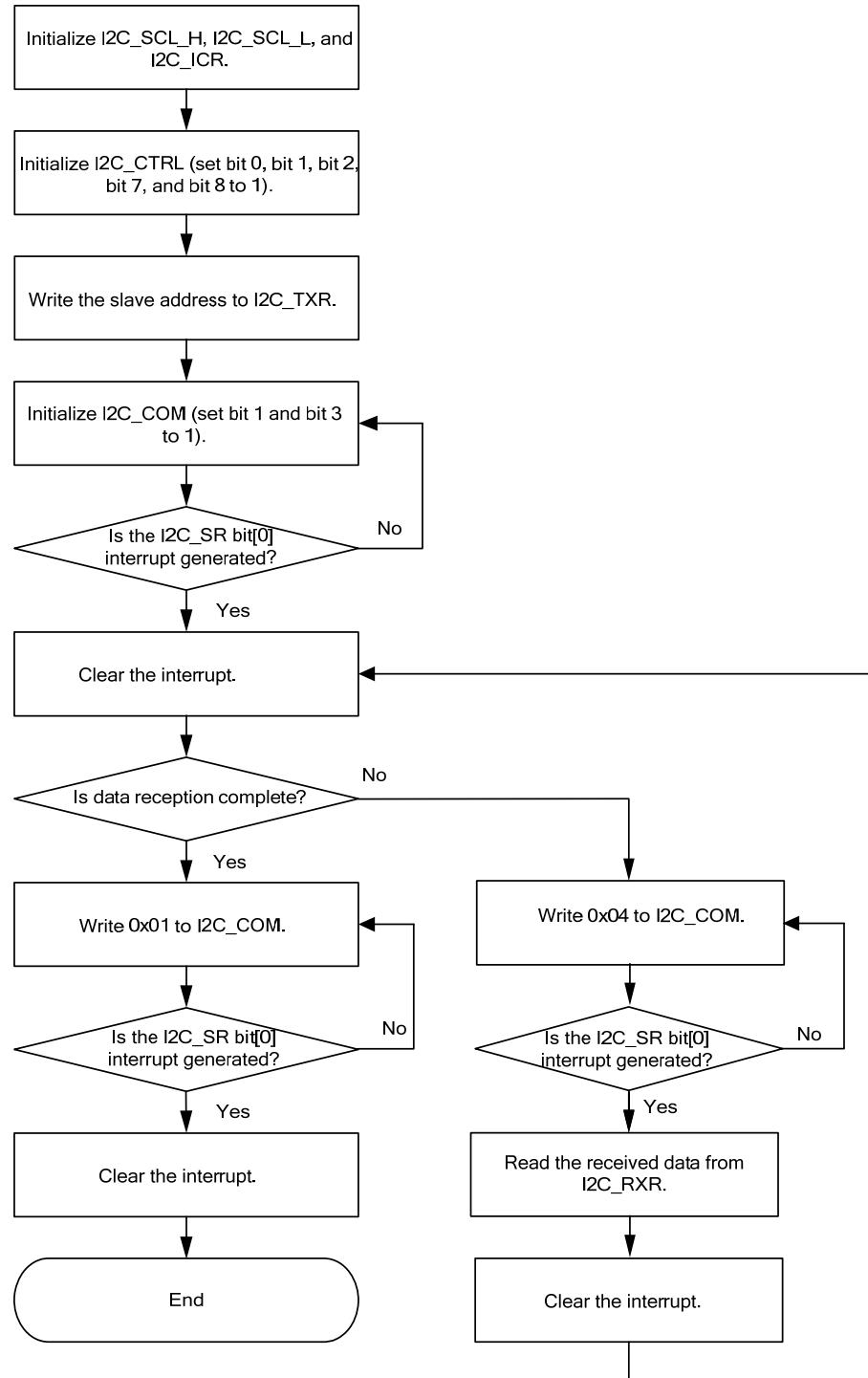
The I²C master can write data to the slave or receive data from the slave. [Figure 11-3](#) shows the process for transmitting data by using the I²C master.

Figure 11-3 Process for transmitting data by using the I²C master

11.3.3.2 Receiving Data by Using the I²C Master

Figure 11-4 shows the process for receiving data by using the I²C master.

Figure 11-4 Process for receiving data by using the I²C master





11.3.4 Register Summary

Hi3796M V100 has three I²C modules. The following are the base addresses for the I²C registers:

- The base address for I²C0 registers is 0xF8B1_0000.
- The base address for I²C1 registers is 0xF8B1_1000.
- The base address for I²C2 registers is 0xF8B1_2000.

Table 11-4 describes I²C registers.

Table 11-4 Summary of I²C registers

Offset Address	Register	Description	Page
0x00	I2C_CTRL	I ² C control register	11-29
0x04	I2C_COM	I ² C command register	11-31
0x08	I2C_ICR	I ² C interrupt clear register	11-32
0x0C	I2C_SR	I ² C module status register	11-33
0x10	I2C_SCL_H	I ² C SCL high-level cycle number register	11-34
0x14	I2C_SCL_L	I ² C SCL low-level cycle number register	11-35
0x18	I2C_TXR	I ² C TX data register	11-35
0x1C	I2C_RXR	I ² C RX data register	11-36

11.3.5 Register Description

I2C_CTRL

I2C_CTRL is an I²C control register. It is used to enable I²C modules and mask interrupts.





[0]	RW	int_done_mask	Bus transfer completion interrupt mask 0: masked 1: not masked
-----	----	---------------	--

I2C_COM

I2C_COM is an I²C command register. It is used to configure the commands for the I²C module.



CAUTION

During or before system initialization, the corresponding interrupts must be cleared.
I2C_COM bit[3:0] are automatically cleared after operations are complete.

	Offset Address 0x04																Register Name I2C_COM								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									op_ack	op_start	op_rd	op_we	op_stop		
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved		Reserved																												
[4]	RW	op_ack		Whether the master sends an ACK as a receiver 0: yes 1: no																												
[3]	RW	op_start		Start condition operation 0: The operation is complete. 1: The operation is valid.																												
[2]	RW	op_rd		Read operation 0: The operation is complete. 1: The operation is valid.																												
[1]	RW	op_we		Write operation 0: The operation is complete. 1: The operation is valid.																												

[0]	RW	op_stop	Stop condition operation 0: The operation is complete. 1: The operation is valid.
-----	----	---------	---

I2C_ICR

I2C_ICR is an I²C interrupt clear register.



CAUTION

When a new interrupt is generated, the I²C module automatically clears the corresponding bit of I2C_ICR.



[2]	WC	clr_int_ack_err	Slave ACK error interrupt clear 0: not cleared 1: cleared
[1]	WC	clr_int_arb_loss	Bus arbitration failure interrupt clear 0: not cleared 1: cleared
[0]	WC	clr_int_done	Bus transfer completion interrupt clear 0: not cleared 1: cleared

I2C_SR

I2C_SR is an I²C module status register. It is used to read the operating status of the I²C module.



CAUTION

I2C_SR bit[1] indicates the I²C bus arbitration failure. When I2C_SR bit[1] is valid, the current operation fails. Before clearing I2C_SR bit[1], you must clear other interrupts, clear I2C_COM or write a new operation command to I2C_COM and then clear I2C_SR bit[1].

	Offset Address 0x0C																Register Name I2C_SR								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved																									bus_busy	int_start	int_stop	int_tx	int_rx	int_ack_err	int_arb_loss	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		
Bits	Access	Name			Description																												
[31:8]	-	reserved			Reserved																												
[7]	RO	bus_busy			Bus busy flag 0: ready 1: busy																												



[6]	RO	int_start	Master start condition TX completion interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	int_stop	Master stop condition TX completion interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	int_tx	Master TX interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.
[3]	RO	int_rx	Master RX interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	int_ack_err	Slave ACK error interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	int_arb_loss	Bus arbitration failure interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	int_done	Bus transfer completion interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.

I2C_SCL_H

I2C_SCL_H is an I²C SCL high-level cycle number register. It is used to configure the number of SCL high-level cycles when the I²C module is working.



CAUTION

During or before system initialization, set [I2C_CTRL](#) bit[7] to 0.



Offset Address																Register Name								Total Reset Value								
0x10																I2C_SCL_H								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																scl_h															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	scl_h				Number of SCL high-level cycles. The actual number of SCL high-level cycles is the configured value multiplied by 2.																										

I2C_SCL_L

I2C_SCL_L is an I²C SCL low-level cycle number register. It is used to configure the number of SCL low-level cycles when the I²C module is working.



CAUTION

During or before system initialization, set [I2C_CTRL](#) bit[7] to 0.

Offset Address																Register Name								Total Reset Value								
0x14																I2C_SCL_L								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																scl_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		
Bits	Access	Name				Description																										
[31:16]	-	reserved				Reserved																										
[15:0]	RW	scl_l				Number of SCL low-level cycles. The actual number of SCL low-level cycles is the configured value multiplied by 2.																										

I2C_TXR

I2C_TXR is an I²C TX data register. It is used to configure the data to be transmitted.



CAUTION

After data transmission is complete, the I²C module does not modify I2C_TXR.

	Offset Address																Register Name								Total Reset Value							
	0x18																I2C_TXR								0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								i2c_txr							
Reset	0	0	0	0	0	0	0	0	0	0	0	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	i2c_txr			Data to be transmitted from the master																											

I2C_RXR

I2C_RXR is an I²C RX data register. It is used to store data received by the master from the slave.

	Offset Address																Register Name								Total Reset Value							
	0x1C																I2C_RXR								0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									i2c_rxr						
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RO	i2c_rxr			Data received by the master																											

11.4 IR

11.4.1 Overview

The IR module receives data over an infrared interface.

11.4.2 Features

The IR module has the following features:

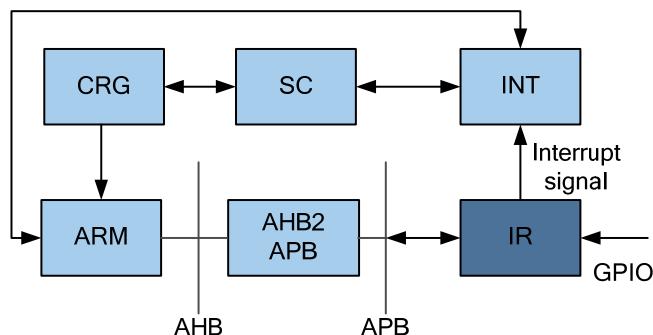
- Allows software to disable the IR module.
- Supports two operating modes:
 - Mode 0 supports RX data error detection, IR wakeup, and decoding in the formats of NEC with simple repeat code, NEC with full repeat code, SONY, and TC9012.
 - Mode 1 supports symbol level width detection in any data format.
- Supports the RX data frame overflow interrupt, RX data frame format error interrupt, RX data frame interrupt, key release interrupt, and combined interrupt in mode 0.
- Supports the RX symbol overflow interrupt, symbol RX interrupt, symbol timeout interrupt, and combined interrupt in mode 1.
- Supports querying of the status of raw and masked interrupts.
- Supports interrupt clear and mask (write to clear).
- Supports IR wakeup.
- Supports the reference clock frequency ranging from 1 MHz to 128 MHz and controls the clock divider by means of software programming, enabling the working clock frequency to be prescaled to 1 MHz.

11.4.3 Function Description

When the IR module receives infrared signals transmitted by an infrared remote control, it decodes the signals and transmits the decoded signals to the ARM subsystem. The ARM subsystem performs corresponding operations based on the received codes to implement the expected function. The IR module connects to the APB in the ARM subsystem.

[Figure 11-5](#) shows the functional block diagram of the IR module.

Figure 11-5 Functional block diagram of the IR module





The analysis on the signals transmitted by various infrared remote controls shows that the lead codes in the infrared commands vary according to remote controls. The subsequent control commands and the number of command code bits are also different. This is because infrared remote controls are not designed based on a unified standard. The basic encoding principles, however, are the same. The pulses with different cycles and duty ratios are used to represent 0 and 1. The duty ratios and pulse cycles may vary according to remote controls. Based on these differences, the code formats of infrared data are classified into the following: NEC with simple repeat code, NEC with full repeat code, TC9012 code, and SONY code.

Table 11-5 to Table 11-7 describe the code formats of the received infrared data.

Table 11-5 Code formats of the received infrared 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
Bit 0 (10 μs)	B0_L	56	56	56	42.2
	B0_H	56	56	56	42.2
Bit 1 (10 μs)	B1_L	56	56	56	42.2
	B1_H	169	169	169	126.6
Simple repeat code (10 μs)	SLEAD_S	900	900	900	337.6
	SLEAD_E	225	225	225	337.6
Burst (10 μs)		55	55	55	42.2
Frame length (10 μs)		10800	10800	10800	8777.6–12828.8
Valid data bit		32	32	42	48

Table 11-6 Code formats of the received infrared data (NEC with full repeat code)

Data Format		NEC with Full Repeat Code						
		uPD6121G	LC7461 M-C13	MN602 4-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
Bit 0 (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)	B1_L	56	56	84.4	87.3	43.6	87.3	88



Data Format		NEC with Full Repeat Code						
		uPD6121G	LC7461 M-C13	MN602 4-C5D6	MN6014 -C6D6	MATNEW	MN6030	PANA SONIC
μs)	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 length (10 μs)		10800	10800	10130	10470	12413.6–16 594.4	10500	10400
Valid data bit		32	42	22	24	48	22	22

Table 11-7 Code formats of the received infrared data (TC9012 code and SONY code)

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 length (10 μs)		10800	4500	4500	4500	4500
Valid data bit		32	12	13	15	20

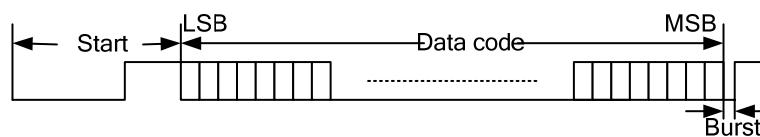
11.4.3.2 NEC with Simple Repeat Code

Frame Format

A frame in NEC with simple repeat code format consists of a start signal (lead code), a data code, and a burst signal. The start signal consists of a start code (low level) and an end code (high level). The number of valid data bits and the meaning of a specific bit in the data code depend on the specific code format, and the LSB of the data code is received first. The burst signal is used to receive the last data bit.

[Figure 11-6](#) shows the format for transmitting a single frame in NEC with simple repeat code format.

Figure 11-6 Format for transmitting a single frame in NEC with simple repeat code format



If a complete data frame is received after a key is held down for more than one frame duration, the subsequent received data frame consists of only a simple lead code and a burst signal. The simple lead code also consists of a start code (low level) and an end code (high level). [Figure 11-7](#) shows the format for transmitting consecutive frames in NEC with simple repeat code format by holding the key down.

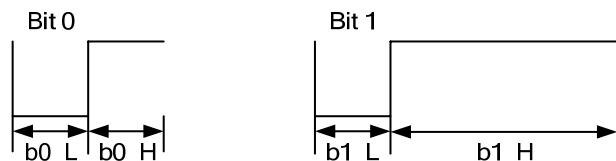
[Figure 11-7](#) Format for transmitting consecutive frames in NEC with simple repeat code format by holding a key down



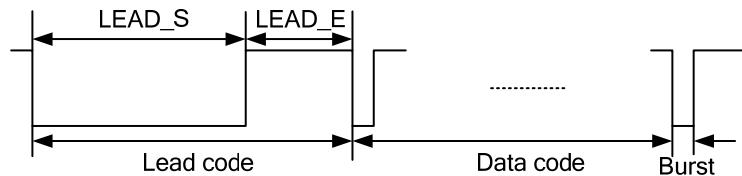
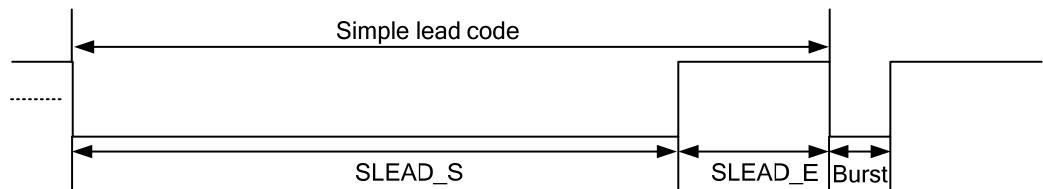
Code Format

[Figure 11-8](#) shows the definitions of bit 0 and bit 1 of the NEC with simple repeat code.

Figure 11-8 Definitions of bit 0 and bit 1 of the NEC with simple repeat code



[Figure 11-9](#) shows the code format for transmitting a single NEC with simple repeat code. [Figure 11-10](#) shows the code format for transmitting consecutive NEC with simple repeat codes by holding the key down.

Figure 11-9 Format for transmitting an NEC with simple repeat code**Figure 11-10** Format for transmitting consecutive NEC with simple repeat codes

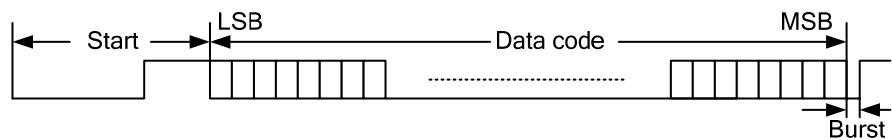
The pulse width of the high and low levels and the frame duration depend on specific code formats. For details, see [Table 11-5](#) to [Table 11-7](#).

The frame duration must be less than or equal to 160 ms. Otherwise, the simple lead code cannot be identified.

11.4.3.3 NEC with Full Repeat Code

Frame Format

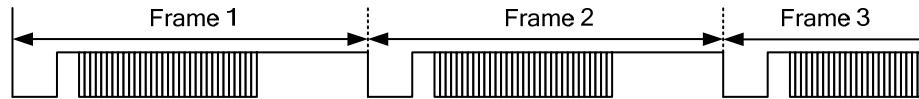
A frame in NEC with full repeat code format consists of a start signal (lead code), a data code, and a burst signal. The start signal consists of a start code (low level) and an end code (high level). The number of valid data bits and the meaning of a specific bit in the data code depend on the specific code format, and the LSB of the data code is received first. The burst signal is used to receive the last data bit. [Figure 11-11](#) shows the format for transmitting a single frame in NEC with full repeat code format.

Figure 11-11 Format for transmitting a single frame in NEC with full repeat code format

If a complete data frame (the first frame) is received after a key is held down for more than one frame duration, the subsequent received data frame is still a complete data frame. That is, the first frame is transmitted repeatedly based on the frame interval. [Figure 11-12](#) shows the format for transmitting consecutive frames in NEC with full repeat code format by holding a key down.



Figure 11-12 Format for transmitting consecutive frames in NEC with full repeat code format by holding a key down

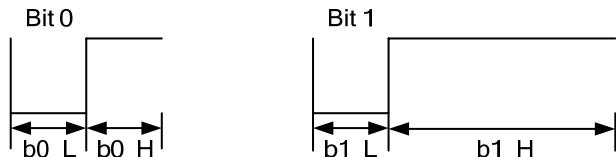


[Figure 11-7](#) and [Figure 11-12](#) show that the only difference between the NEC with simple repeat code and the NEC with full repeat code is the format of the repeat frame. For the NEC with simple repeat code, a simple lead code is transmitted; for the NEC with full repeat code, complete frames are transmitted. The repeat frame is the same as the first frame.

Code Format

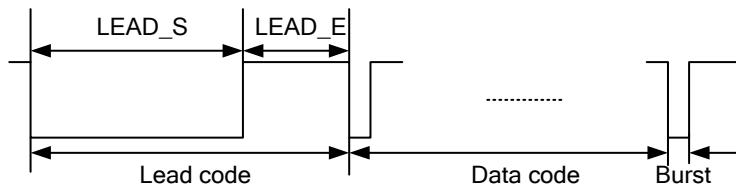
[Figure 11-13](#) shows the definitions of bit 0 and bit 1 of the NEC with full repeat code.

Figure 11-13 Definitions of bit 0 and bit 1 of the NEC with full repeat code



[Figure 11-14](#) shows the format for transmitting a single NEC with full repeat code.

Figure 11-14 Format for transmitting a single NEC with full repeat code



The pulse width of the high and low levels and the frame duration depend on specific code formats. For details, see [Table 11-5](#) to [Table 11-7](#).

11.4.3.4 TC9012

Frame Format

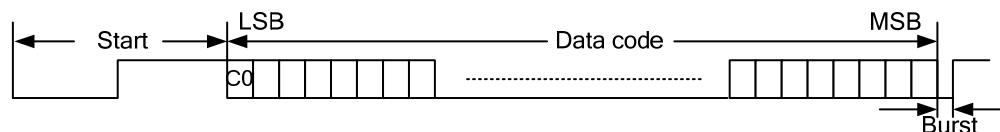


CAUTION

According to the TC9012 code features, the first bit of all key codes must be 1 or 0. Otherwise, unexpected frames are generated when a key is held down.

A frame in TC9012 code format consists of a start signal (lead code), a data code, and a burst signal. The start signal consists of a start code (low level) and an end code (high level). The number of valid data bits and the meaning of a specific bit in the data code depend on the specific code format, and the LSB of the data code is received first. The burst signal is used to receive the last data bit. Figure 11-15 shows the format for transmitting a single frame in TC9012 code format.

Figure 11-15 Format for transmitting a single frame in TC9012 code format



If a complete data frame is received after a key is held down for more than one frame duration, the subsequent received data frame consists of a lead code, a data bit, and a burst signal. The lead code also consists of a start code (low level) and an end code (high level). The data bit is the inverse code of the first data bit (C_0) received in the previous frame. Figure 11-16 shows the format for transmitting consecutive frames in TC9012 code format by holding a key down.

Figure 11-16 Format for transmitting consecutive frames in TC9012 code format by holding a key down



Code Format

Figure 11-17 shows the definitions of bit 0 and bit 1 of the TC9012 code.

Figure 11-17 Definitions of bit 0 and bit 1 of the TC9012 code

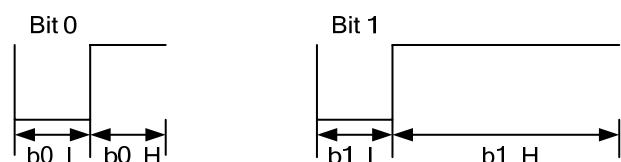


Figure 11-18 shows the format for transmitting a single TC9012 code.

Figure 11-18 Format for transmitting a single TC9012 code

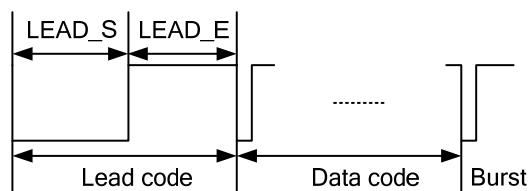


Figure 11-19 shows the format for transmitting consecutive TC9012 codes when C0 is 1.

Figure 11-19 Format for transmitting consecutive TC9012 codes when C0 is 1

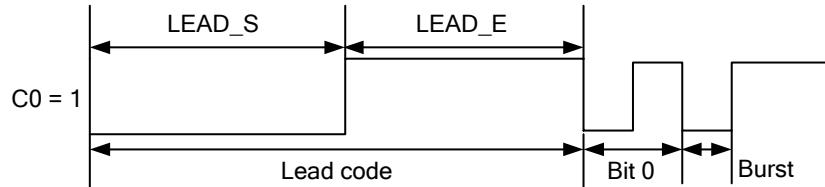
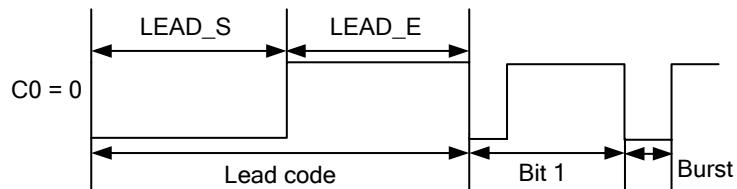


Figure 11-20 shows the format for transmitting consecutive TC9012 codes when C0 is 0.

Figure 11-20 Format for transmitting consecutive TC9012 codes when C0 is 0



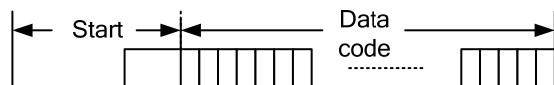
Note: The pulse width of the high and low levels and the frame duration depend on specific code formats. For details, see [Table 11-5](#) to [Table 11-7](#). The frame duration must be less than or equal to 160 ms. Otherwise, the repeat frame cannot be identified.

11.4.3.5 SONY

Frame Format

A frame in SONY format consists of a start signal (lead code) and a data code. The start signal consists of a start code (low level) and an end code (high level). The number of valid data bits and the meaning of a specific bit in the data code depend on the specific code format, and the LSB of the data code is received first. Figure 11-21 shows the format for transmitting a single frame in SONY code format.

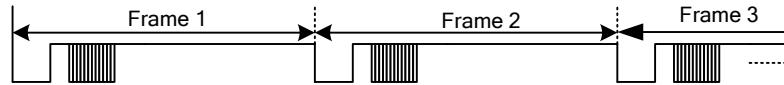
Figure 11-21 Format for transmitting a single frame in SONY code format



If a complete data frame is received after a key is held down for more than one frame duration, the subsequent received data frame is also a complete data frame. Figure 11-22 shows the format for transmitting consecutive frames in SONY code format by holding a key down.



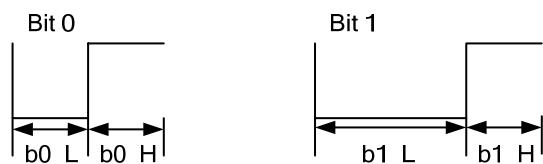
Figure 11-22 Format for transmitting consecutive frames in SONY code format by holding a key down



Bit Format

Figure 11-23 shows the definitions of bit 0 and bit 1 of the SONY code.

Figure 11-23 Definitions of bit 0 and bit 1 of the SONY code



Note: The pulse width of the high and low levels and the frame duration depend on specific code formats. For details, see [Table 11-5](#) to [Table 11-7](#).

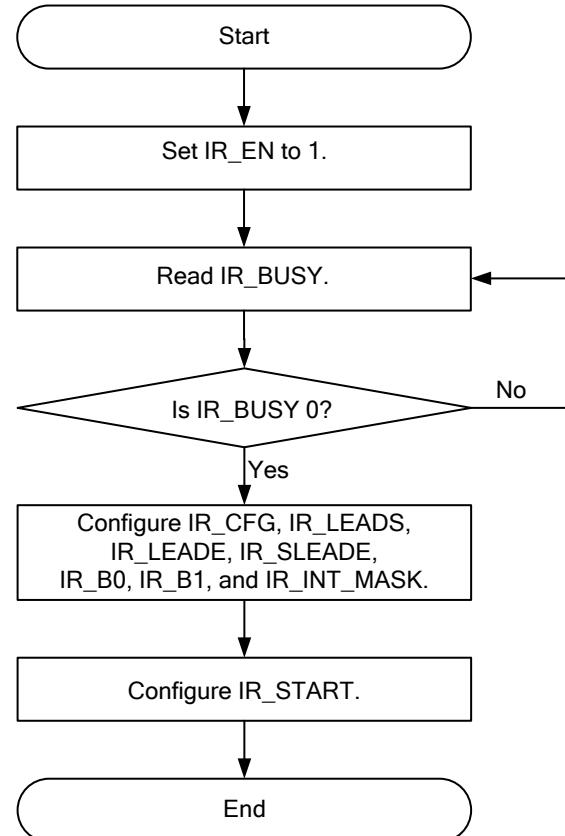
11.4.4 Operating Mode

Soft Reset

The IR module can be separately soft-reset by setting SC_CLKGATE_SRST_CTRL[ir_srst_req] to 1. After reset is deasserted, the configuration registers are restored to default values. Therefore, these registers must be reinitialized after reset.

Register Configuration Instances

Figure 11-24 shows the process for initializing the IR module.

Figure 11-24 Process for initializing the IR module

To initialize the IR module, perform the following steps:

Step 1 Select the address space of the IR module.

Step 2 Set [IR_EN](#) bit[0] to 1 to enable the IR RX module.

Step 3 Read [IR_BUSY](#) to check the current status of the IR module.

- If the read value is 1, the IR module is busy. Then continue to read [IR_BUSY](#). Note that configuring other control registers of the IR module by using software has no effect at this time.
- If the read value is 0, the IR module is idle. Then go to step 4.

Step 4 Configure [IR_CFG](#), [IR_LEADS](#), [IR_LEADE](#), [IR_SLEADE](#), [IR_B0](#), [IR_B1](#), and [IR_INT_MASK](#).

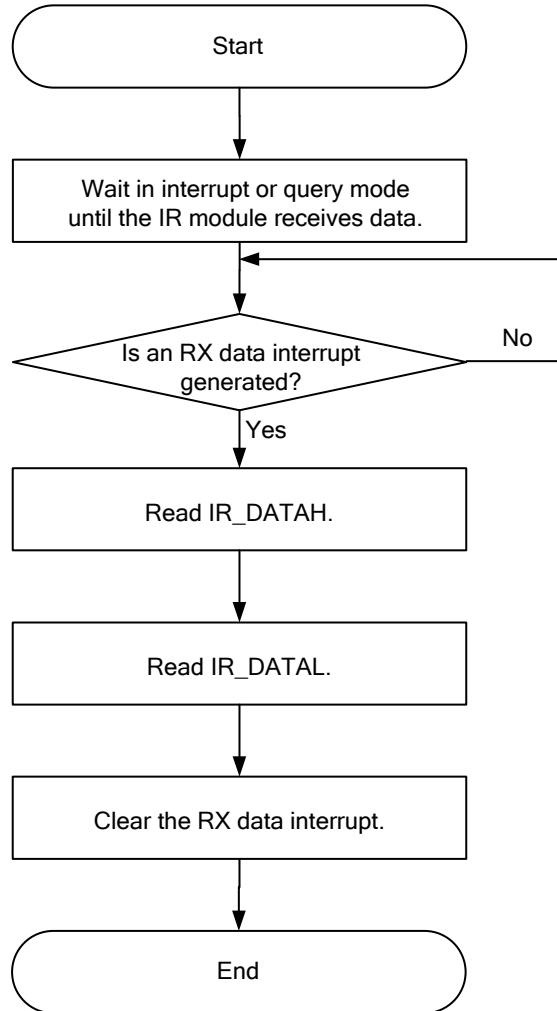
Note: You can update corresponding registers as required. If the registers are not updated, the original values are retained.

Step 5 Configure [IR_START](#) after all IR control registers are configured. This is because [IR_START](#) is used to generate the start signal. If [IR_START](#) is configured, the IR module starts to receive infrared data based on the values of IR control registers.

----End



Figure 11-25 Process for reading the decoded data



To read the decoded data, perform the following steps:

Step 1 Initialize the IR module.

Step 2 Wait in interrupt or query mode until data frames are received.

- In interrupt mode, the CPU queries `IR_INT_STATUS[intms_rcv]` when it receives an interrupt request signal from the IR module. If the read value is 1, the IR module receives a data frame. Then go to step 3. If the read value is 0, repeat step 2 to wait for an interrupt.
- In query mode, the software continuously or periodically reads `IR_INT_STATUS[intrs_rcv]`. If the read value is 1, the IR module receives a data frame. Then go to step 3. If the read value is 0, the IR module does not receive a data frame. Then repeat step 2 to continue the query.

Step 3 Read `IR_DATAH`. Skip this step if data in a frame does not exceed 32 bits.

Step 4 Read `IR_DATAL`.

Step 5 Clear the RX data interrupt.

----End



11.4.5 Register Summary

Table 11-8 describes IR registers.

Table 11-8 Summary of IR registers (base address: 0xF8001000)

Offset Address	Register	Description	Page
0x000	IR_EN	IR RX enable control register	11-49
0x004	IR_CFG	IR configuration register	11-49
0x008	IR_LEADS	Lead code start bit margin configuration register (valid only when IR_CFG[ir_mode] is 0)	11-51
0x00C	IR_LEADE	Lead code end bit margin configuration register (valid only when IR_CFG[ir_mode] is 0)	11-52
0x010	IR_SLEADE	Simple lead code end bit margin configuration register (valid only when IR_CFG[ir_mode] is 0)	11-53
0x014	IR_B0	Data 0 level judge margin configuration register (valid only when IR_CFG[ir_mode] is 0)	11-54
0x018	IR_B1	Data 1 level judge margin configuration register (valid only when IR_CFG[ir_mode] is 0)	11-55
0x01C	IR_BUSY	Configuration busy flag register	11-56
0x020	IR_DATAH	Upper 16-bit IR RX decoded data register (when IR_CFG[ir_mode] is 0) or symbol FIFO symbol count register (when IR_CFG[ir_mode] is 1)	11-56
0x024	IR_DATAL	Lower 32-bit IR RX decoded data register (when IR_CFG[ir_mode] is 0) or IR received symbol width register (when IR_CFG[ir_mode] is 1)	11-57
0x028	IR_INT_MASK	IR interrupt mask register	11-58
0x02C	IR_INT_STATUS	IR interrupt status register	11-60
0x030	IR_INT_CLR	IR interrupt clear register	11-62
0x034	IR_START	IR start configuration register	11-64



11.4.6 Register Description

IR_EN

IR_EN is an IR RX enable control register.



CAUTION

Before configuring other registers, set IR_EN[ir_en] to 1 by using software. Otherwise, configuring other registers has no effect. When IR_EN[ir_en] is 0, 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	0	0	0	0	0	0	0	0								
Bits	Access	Name		Description																																												
[31:1]	-	reserved		Reserved																																												
[0]	RW	ir_en		IR RX enable 0: disabled 1: enabled																																												

IR_CFG

IR_CFG is an IR configuration register.



CAUTION

Before setting this register, ensure that IR_BUSY[ir_busy] is set to 0 and IR_EN[ir_en] is set to 1. Otherwise, the original value is retained after setting.

The reference clock frequency supported by the IR module ranges from 1 MHz to 128 MHz. The following describes the relationship between the frequency and the clock divider ir_freq:

- When the reference clock frequency is 1 MHz, ir_freq must be set to 0x00.
- When the reference clock frequency is 128 MHz, ir_freq must be set to 0x7F.



When the frequency of the IR reference clock ranging from 1 MHz to 128 MHz is not an integer, the clock divider is selected based on the rounded reference clock frequency. For example, if the reference clock frequency is 12.1 MHz, the clock divider is 0x0B; if the reference clock frequency is 12.8 MHz, the clock divider is 0x0C.

The following describes the relationship between the frequency offset and the count deviation. If the base frequency is f and the frequency variation is D_f , the frequency offset ratio is D_f/f . If the count deviation of the counter is D_{cnt} , and the judge level width is s (in μs), the count deviation D_{cnt} is calculated as follows: $D_{cnt} = 0.1 \times s \times \text{ratio}$. Therefore, when the clock has frequency offset, the valid range of the parameter value needs to be shifted. If the frequency increases, the corresponding margin range is changed to $[\min + D_{cnt}, \max + D_{cnt}]$, where \min and \max indicate the margins without frequency offset. If the frequency decreases, the offset range is changed to $[\min - D_{cnt}, \max - D_{cnt}]$. Take the margin of the start bit in the lead code as an example. If the base frequency is 100 MHz, and the frequency increases by 0.1 MHz, the ratio is 0.001 (0.1/100). If s is 9000 μs , D_{cnt} is calculated as follows: $D_{cnt} = 0.1 \times 9000 \times 0.001 = 1$. In this case, the margin range of `ir_leads` must be changed to [0x033D, 0x3CD].

	Offset Address	Register Name	Total Reset Value		
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	IR_CFG	0x3E80_1F0B		
Name	ir_max_level_width	ir_format	ir_bits	ir_mode	ir_freq
Reset	0 0 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 0 1 1				
Bits	Access	Name	Description		
[31:16]	RW	ir_max_level_width	Invalid when IR_CFG[ir_mode] is 0 Maximum level width (in 10 μs) of a symbol when IR_CFG[ir_mode] is 1. This width identifies the end of a symbol stream.		
[15:14]	RW	ir_format	Data code format when IR_CFG[ir_mode] is 0 00: NEC with simple repeat code 01: TC9012 code 10: NEC with full repeat code 11: SONY code For details about the relationship between code types and code formats, see Table 11-5 to Table 11-7 . Symbol format when IR_CFG[ir_mode] is 1 Bit[15]: reserved Bit[14]: 0: The symbol is from low to high, and the symbol stream ends at the high level. 1: The symbol is from high to low, and the symbol stream ends at the low level.		



[13:8]	RW	ir_bits	Number of data bits in a frame when IR_CFG[ir_mode] is 0 0x00–0x2F: correspond to bit 1 to bit 48 in a frame respectively 0x30–0x3F: reserved If ir_bits is set to a value ranging from 0x30 to 0x3F by using software, the setting has no effect and the original value is retained. Symbol RX interrupt threshold when IR_CFG[ir_mode] is 1 Bit[13:8]: 0x0–0x3F. 0x0 indicates that an interrupt is reported when there is at least one symbol in the FIFO, 0x3F indicates that an interrupt is reported when there are at least 64 symbols in the FIFO, and so on.
[7]	RW	ir_mode	IR operating mode 0: The decoded complete data frames are output. 1: Only the symbol width is output.
[6:0]	RW	ir_freq	Working clock divider 0x00–0x7F correspond to the clock divider 1–128 respectively.

IR_LEADS

IR_LEADS is a lead code start bit margin configuration register (valid only when [IR_CFG\[ir_mode\]](#) is 0).



CAUTION

Before setting this register, ensure that [IR_BUSY\[ir_busy\]](#) is set to 0 and [IR_EN\[ir_en\]](#) is set to 1. Otherwise, the original value is retained after setting.

The margin must be considered based on the typical value of the specific code type for accurately determining the start bit of the lead code. For details about the typical values of specified code types, see the values of LEAD_S in [Table 11-5](#) to [Table 11-7](#).

- For the pulse width whose typical value is greater than or equal to 400 (10 μ s precision), the recommended margin is 8% of the typical value. For example, if the typical value of LEAD_S for D6121 is 900, the value of cnt_leads_min is 0x33C ($900 \times 92\% = 828$), and the value of cnt_leads_max is 0x3CC ($900 \times 108\% = 972$).
- For the pulse width whose typical value is less than 400 (10 μ s precision), the recommended margin is 20% of the typical value. For example, if the typical value of LEAD_S for SONY-D7C5 is 240, the value of cnt_leads_min is 0xC0 ($240 \times 80\% = 192$), and the value of cnt_leads_max is 0x120 ($240 \times 120\% = 288$).

The basic configuration principles are as follows: cnt_leads_max is greater than or equal to 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	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	0	
Bits	Access	Name		Description																												
[31:26]	-	reserved		Reserved																												
[25:16]	RW	cnt_leads_min		Minimum pulse width of the start bit of the lead code 0x000–0x007: reserved																												
[15:10]	-	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 a lead code end bit margin configuration register (valid only when [IR_CFG\[ir_mode\]](#) is 0).



CAUTION

- Before configuring this register, ensure that IR_BUSY[ir_busy] is set to 0 and IR_EN[ir_en] is set to 1. Otherwise, the original value is retained after configuration.
- For the NEC with simple repeat codes, the margins of cnt_sleade and cnt_leade cannot overlap. Otherwise, when the actual count value falls within the overlap range, the simple lead code cannot be identified. As a result, a frame format error occurs.

The margin must be considered based on the typical value of the specific code type for accurately determining the end bit of the lead code. The margin is about 8% of the typical value. For details about the typical values of specific code types, see the values of LEAD_E in [Table 11-5](#) to [Table 11-7](#).

- For the pulse width whose typical value is greater than or equal to 400 (10 µs precision), the recommended margin is 8% of the typical value. For example, if the typical value of LEAD_E for D6121 is 450, the value of cnt_leade_min is 0x19E ($450 \times 92\% = 414$), and the value of cnt_leade_max is 0x1E6 ($450 \times 108\% = 486$).
- For the pulse width whose typical value is less than 400 (10 µs precision), the recommended margin is 20% of the typical value. For example, if the typical value of LEAD_E for SONY-D7C5 is 60, the value of cnt_leade_min is 0x030 ($60 \times 80\% = 48$), and the value of cnt_leade_max is 0x048 ($60 \times 120\% = 72$).

The basic configuration principle is that cnt_leade_max must be greater than or equal to 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	1	1	1	1	0	0	1	1	0	
Bits	Access	Name		Description																												
[31:25]	-	reserved		Reserved																												
[24:16]	RW	cnt_leade_min		Minimum pulse width of the end bit of the lead code 0x000–0x007: reserved																												
[15:9]	-	reserved		Reserved																												
[8:0]	RW	cnt_leade_max		Maximum pulse width of the end bit of the lead code 0x000–0x007: reserved																												

IR_SLEADE

IR_SLEADE is a simple lead code end bit margin configuration register (valid only when [IR_CFG\[ir_mode\]](#) is 0).



CAUTION

- Before setting this register, ensure that IR_BUSY[ir_busy] is set to 0 and IR_EN[ir_en] is set to 1. Otherwise, the original value is retained after setting.
- For the NEC with simple repeat codes, the margins of cnt_sleade and cnt_leade cannot overlap. Otherwise, when the actual count value falls within the overlap range, the simple lead code cannot be identified. As a result, a frame format error occurs.
- This register needs to be configured only for the NEC with simple repeat code.

The margin must be considered based on the typical value of the specific code type for accurately determining the end bit of the simple lead code.

For details about the typical values of specific code types, see the values of SLEAD_E in [Table 11-5](#) to [Table 11-7](#).

- For a pulse width whose typical value is greater than or equal to 225 (10 µs precision), the recommended margin is 8% of the typical value. For example, if the typical value of SLEAD_E for D6121 is 225, the value of cnt_sleade_min is 0xCF (225 x 92% = 207), and the value of cnt_sleade_max is 0xF3 (225 x 108% = 243). For a pulse width whose typical value is less than 225 (10 µs precision), the recommended margin is 20% of the typical value. For example, if the typical value of SLEAD_E for a code is 60, the value of cnt_sleade_min is 0x030 (60 x 80% = 48), and the value of cnt_sleade_max is 0x048 (60 x 120% = 72).



The basic configuration principle is that cnt_sleade_max must be greater than or equal to 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	1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1
Bits	Access	Name				Description																										
[31:25]	-	reserved				Reserved																										
[24:16]	RW	cnt_sleade_min				Minimum pulse width of the end bit of the simple lead code 0x000–0x007: reserved																										
[15:9]	-	reserved				Reserved																										
[8:0]	RW	cnt_sleade_max				Maximum pulse width of the start bit of the simple lead code 0x000–0x007: reserved																										

IR_B0

IR_B0 is a data 0 level judge margin configuration register (valid only when IR_CFG[ir_mode] is 0).



CAUTION

- Before setting this register, ensure that IR_BUSY[ir_busy] is set to 0 and set IR_EN[ir_en] is set to 1. Otherwise, the original value is retained after setting.
- The level judge margin ranges for bit 0 and bit 1 of the four types of codes cannot overlap. Otherwise, when the actual count value falls within the overlap range, bit 1 cannot be identified and is regarded as bit 0 by mistake.

The margin must be considered based on the typical value of the specific code type for accurately determining bit 0. 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 B0_H in [Table 11-5](#) to [Table 11-7](#). For example, if the typical value of B0_H for D6121 is 56 (10 µs precision), cnt0_b_min is 0x2D ($56 \times 80\% = 45$), and cnt0_b_max is 0x43 ($56 \times 120\% = 67$).
- For details about the typical values of SONY codes, see the values of B0_L in [Table 11-5](#) to [Table 11-7](#). For example, if the typical value of B0_L for SONY-D7C5 is 60 (10 µs precision), cnt0_b_min is 0x30 ($60 \times 80\% = 48$), and cnt0_b_max is 0x48 ($60 \times 120\% = 72$).



The basic configuration principle is that cnt0_b_max must be greater than or equal to 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	0	1	0	0	0	1	1
Bits	Access	Name			Description																											
[31:25]	-	reserved			Reserved																											
[24:16]	RW	cnt0_b_min			Minimum pulse width of the level for determining bit 0 0x000–0x007: reserved																											
[15:9]	-	reserved			Reserved																											
[8:0]	RW	cnt0_b_max			Maximum pulse width of the level for determining bit 0 0x000–0x007: reserved																											

IR_B1

IR_B1 is a data 1 judge level margin configuration register (valid only when IR_CFG[ir_mode] is 0).



CAUTION

- Before configuring this register, ensure that IR_BUSY[ir_busy] is set to 0 and IR_EN[ir_en] is set to 1. Otherwise, the original value is retained after configuration.
- The level judge margin ranges for bit 0 and bit 1 of the four types of codes cannot overlap. Otherwise, when the actual count value falls within the overlap range, bit 1 cannot be identified and is regarded as bit 0 by mistake.

The margin must be considered based on the typical value of the specific code type for accurately determining bit 1. 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 11-5](#) to [Table 11-7](#). For example, if the typical value of B1_H for D6121 is 169 (10 µs precision), cnt1_b_min is 0x87 ($169 \times 80\% = 135$), and cnt1_b_max is 0xCB ($169 \times 120\% = 203$).
- For details about the typical values of SONY codes, see the values of B1_L in [Table 11-5](#) to [Table 11-7](#). For example, if the typical value of B1_L for SONY-D7C5 is 120 (10 µs precision), cnt1_b_min is 0x60 ($120 \times 80\% = 96$), and cnt1_b_max is 0x90 ($120 \times 120\% = 144$).

The basic configuration principle is that cnt1_b_max must be greater than or equal to 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	1	1	0	0	1	0	1	1	
Bits	Access	Name			Description																											
[31:25]	-	reserved			Reserved																											
[24:16]	RW	cnt1_b_min			Minimum pulse width of the level for determining bit 1 0x000–0x007: reserved																											
[15:9]	-	reserved			Reserved																											
[8:0]	RW	cnt1_b_max			Maximum pulse width of the level for determining bit 1 0x000–0x007: reserved																											

IR_BUSY

IR_BUSY is a configuration busy flag register.

IR_DATAH

IR_DATAH is an upper 16-bit IR RX decoded data register (when [IR_CFG\[ir_mode\]](#) is 0) or symbol FIFO symbol count register (when [IR_CFG\[ir_mode\]](#) is 1).

When `IR_CFG[ir_mode]` is set to 0, `IR_DATAH` receives the upper 16 bits of the decoded data, whereas `IR_DATAL` receives the lower 32 bits of the decoded data. The valid data bits



depend on the number of valid data bits in a frame of specific code. For details, see the valid data bits in [Table 11-5](#) to [Table 11-7](#).

Data is stored in [IR_DATAL](#) and then [IR_DATAH](#) from LSB to MSB. That is, after [IR_DATAL](#) is full, the remaining data is stored in [IR_DATAH](#). The unused upper bits are reserved. Software must read [IR_DATAH](#) before [IR_DATAL](#).

When [IR_CFG\[ir_mode\]](#) is 1, hardware receives all data bits without considering the definition of each data bit. Then software processes the data bits in a unified manner.

	Offset Address	Register Name	Total Reset Value					
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	IR_DATAH	0x0000_0000					
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]	-	reserved	Reserved					
[15:0]	RO	ir_datah	Upper 16 bits of the decoded data received by the IR module when IR_CFG[ir_mode] is 0 Number of symbols in the symbol FIFO when IR_CFG[ir_mode] is 1 Bit[15:7]: reserved Bit[6:0]: number of symbols in the symbol FIFO					

IR_DATAL

[IR_DATAL](#) is a lower 32-bit IR RX decoded data register (when [IR_CFG\[ir_mode\]](#) is 0) or IR received symbol width register (when [IR_CFG\[ir_mode\]](#) is 1).



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	
Bits	Access	Name	Description																													
[31:0]	RO	ir_datal	<p>Lower 32 bits of the decoded data received by the IR module when IR_CFG[ir_mode] is 0</p> <p>Width of the symbol received by the IR module when IR_CFG[ir_mode] is 1</p> <p>Bit[31:16]:</p> <p>When the symbol is low and then high, these bits indicate the high-level width (in 10 µs) of the symbol received by the IR module.</p> <p>When the symbol is high and then low, these bits indicate the low-level width (in 10 µs) of the symbol received by the IR module.</p> <p>Bit[15:0]:</p> <p>When the symbol is low and then high, these bits indicate the low-level width (in 10 µs) of the symbol received by the IR module.</p> <p>When the symbol is high and then low, these bits indicate the high-level width (in 10 µs) of the symbol received by the IR module.</p>																													

IR_INT_MASK

IR_INT_MASK is an IR interrupt mask register.



CAUTION

- Before setting this register, ensure that [IR_EN\[ir_en\]](#) is set to 1. Otherwise, the original value is retained after setting.
- If all interrupts are masked, the IR wakeup function is unavailable.
- When [IR_CFG\[ir_mode\]](#) is 0, IR_INT_MASK bit[3:0] are valid; when [IR_CFG\[ir_mode\]](#) is 1, IR_INT_MASK bit[18:16] are valid.

The following describes related interrupts:

- RX data overflow interrupt
If the CPU does not fetch the current frame in time and the subsequent frame is received, the subsequent frame overwrites the current frame, and a raw RX data overflow interrupt is reported.
- RX data frame format error interrupt



If the received data frame is incomplete or the data pulse width does not fall within the margin range, a raw RX frame format error interrupt is reported.

- Data frame RX interrupt

When a complete frame is received, a raw data frame RX interrupt is reported.

- Key release detection interrupt

For the NEC with simple repeat code and TC9012 code data formats, if the start sync code is not detected again within 160 ms after the previous start sync code is detected, or a valid data frame rather than a simple lead code is detected, a raw key release detection interrupt is reported. The key release detection interrupt is not supported for the NEC with full repeat code and the SONY code.

- RX symbol overflow interrupt

If the symbol FIFO is full because the CPU does not fetch the data in time and the subsequent symbol is already received, a raw RX symbol overflow interrupt is reported.

- Symbol RX interrupt

If a complete symbol is received and the number of symbols in the symbol FIFO is above the threshold specified in [IR_CFG\[ir_bits\]](#), a raw symbol RX interrupt is reported.

- Symbol timeout interrupt

If no symbol is received during the period configured in [IR_CFG\[ir_max_level_width\]](#) after a valid symbol is received, a raw symbol timeout interrupt is reported.

Hardware does not identify the interrupt priority. An interrupt is generated if one or more masked interrupt sources are valid.

	Offset Address 0x028																Register Name IR_INT_MASK								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	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								intm_release				intm_overflow			
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved			Reserved																											
[18]	RW	intm_overrun			Symbol overflow interrupt mask when IR_CFG[ir_mode] is 1 0: not masked 1: masked																											
[17]	RW	intm_time_out			Symbol timeout interrupt mask when IR_CFG[ir_mode] is 1 0: not masked 1: masked																											



[16]	RW	intm_symb_rcv	<i>n</i> symbol RX interrupt mask when IR_CFG[ir_mode] is 1 0: not masked 1: masked
[15:4]	-	reserved	Reserved
[3]	RW	intm_release	Key release interrupt mask when IR_CFG[ir_mode] is 0 0: not masked 1: masked
[2]	RW	intm_overflow	RX data overflow interrupt mask when IR_CFG[ir_mode] is 0 0: not masked 1: masked
[1]	RW	intm_framerr	RX data frame format error interrupt mask when IR_CFG[ir_mode] is 0 0: not masked 1: masked
[0]	RW	intm_rcv	Data frame RX interrupt mask when IR_CFG[ir_mode] is 0 0: not masked 1: masked

IR_INT_STATUS

IR_INT_STATUS is an IR interrupt status register.



CAUTION

- When [IR_CFG\[ir_mode\]](#) is 0, IR_INT_STATUS bit[3:0] and IR_INT_STATUS bit[19:16] are valid.
- When [IR_CFG\[ir_mode\]](#) is 1, IR_INT_STATUS bit[10:8] and IR_INT_STATUS bit[26:24] are valid.



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_overrun	intms_time_out	intms_symb_rcv	reserved				intms_release	intms_overflow	intms_framerr	intms_rcv	reserved				intrs_overrun	intrs_time_out	intrs_symb_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					
Bits	Access	Name			Description																														
[31:27]	-	reserved			Reserved																														
[26]	RO	intms_overrun			Masked symbol overflow interrupt status when IR_CFG[ir_mode] is 1 0: No interrupt is generated. 1: An interrupt is generated.																														
[25]	RO	intms_time_out			Masked symbol timeout interrupt status when IR_CFG[ir_mode] is 1 0: No interrupt is generated. 1: An interrupt is generated.																														
[24]	RO	intms_symb_rcv			Masked symbol RX interrupt status when IR_CFG[ir_mode] is 1 0: No interrupt is generated. 1: An interrupt is generated.																														
[23:20]	-	reserved			Reserved																														
[19]	RO	intms_release			Masked key release interrupt status when IR_CFG[ir_mode] is 0 0: No interrupt is generated. 1: An interrupt is generated.																														
[18]	RO	intms_overflow			Masked RX data overflow interrupt status when IR_CFG[ir_mode] is 0 0: No interrupt is generated. 1: An interrupt is generated.																														
[17]	RO	intms_framerr			Masked RX data frame format error interrupt status when IR_CFG[ir_mode] is 0 0: No interrupt is generated. 1: An interrupt is generated.																														



[16]	RO	intms_rcv	Masked data frame RX interrupt status when IR_CFG[ir_mode] is 0 0: No interrupt is generated. 1: An interrupt is generated.
[15:11]	-	reserved	Reserved
[10]	RO	intrs_overrun	Raw symbol overflow interrupt status when IR_CFG[ir_mode] is 1 0: No interrupt is generated. 1: An interrupt is generated.
[9]	RO	intrs_time_out	Raw symbol timeout interrupt status when IR_CFG[ir_mode] is 1 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	intrs_symb_rcv	Raw symbol RX interrupt status when IR_CFG[ir_mode] is 1 0: No interrupt is generated. 1: An interrupt is generated.
[7:4]	-	reserved	Reserved
[3]	RO	intrs_release	Raw key release interrupt status when IR_CFG[ir_mode] is 0 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	intrs_overflow	Raw RX data overflow interrupt status when IR_CFG[ir_mode] is 0 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	intrs_framerr	Raw RX data frame format error interrupt status when IR_CFG[ir_mode] is 0 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	intrs_rcv	Raw data frame RX interrupt status when IR_CFG[ir_mode] is 0 0: No interrupt is generated. 1: An interrupt is generated.

IR_INT_CLR

IR_INT_CLR is an IR interrupt clear register.



CAUTION

- When [IR_CFG\[ir_mode\]](#) is 0, IR_INT_CLR bit[3:0] are valid.



- When IR_CFG[ir_mode] is 1, IR_INT_CLR bit[18:16] are valid.

Offset Address		Register Name																Total Reset Value																
Bit	0x030	IR_INT_CLR																0x0000_0000																
Name	31 30 29 28 27 26 25 24 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																															
[31:19]	-	reserved	Reserved																															
[18]	WC	intc_overrun	Symbol overflow interrupt clear when IR_CFG[ir_mode] is 1 0: not cleared 1: cleared																															
[17]	WC	intc_time_out	Symbol timeout interrupt clear when IR_CFG[ir_mode] is 1 0: not cleared 1: cleared																															
[16]	WC	intc_symb_rcv	Symbol RX interrupt clear when IR_CFG[ir_mode] is 1 0: not cleared 1: cleared																															
[15:4]	-	reserved	Reserved																															
[3]	WC	intc_release	Key release interrupt clear when IR_CFG[ir_mode] is 1 0: not cleared 1: cleared																															
[2]	WC	intc_overflow	RX data overflow interrupt clear when IR_CFG[ir_mode] is 0 0: not cleared 1: cleared																															
[1]	WC	intc_framerr	RX data frame format error interrupt clear when IR_CFG[ir_mode] is 0 0: not cleared 1: cleared																															



[0]	WC	intc_rcv	Data frame RX interrupt clear when IR_CFG[ir_mode] is 0 0: not cleared 1: cleared If software writes 1 to the bit without reading data from IR_DATA after a data frame RX interrupt is generated, the interrupt cannot be cleared.
-----	----	----------	---

IR_START

IR_START is an IR start configuration register.

After other registers are configured, the IR module can be started when any value is written to the corresponding address for IR_START.

	Offset Address 0x034																Register Name IR_START																Total Reset Value 0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	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]	-	reserved			Reserved																																											
[0]	WO	ir_start			IR start configuration register																																											

11.5 LED

11.5.1 Overview

The LED module supports only the CT1642 chip and provides the following functions:

- LED display control. The LED module receives the data to be displayed from the CPU over the APB and transmits the data and CSs to the CT1642 chip in serial mode for display.
- Only 8 x 1 keypad scanning

The working reference clock of the LED module is 24 MHz.

11.5.2 Function Description

The LED module has the following features:



- Integrates an APB slave interface for supporting standard APB operations.
- Supports synchronous dynamic display of a maximum of four LEDs.
- Supports the configurable refresh rate.
- Supports blinking through hardware at a configurable blinking frequency.
- Connects to the CT1642 control chip.
- Implements 8 x 1 matrix keypad scanning.
- Soft-resets the LEDC. The reset signal can be controlled by configuring SC_CLKGATE_SRST_CTRL[led_srst_req].

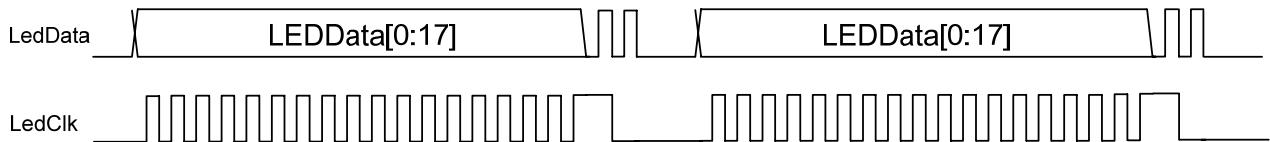
11.5.3 Operating Mode

11.5.3.1 Output Timing of the LED Module

Figure 11-26 shows the output timing of the LED module. LedData and LedClk compose serial synchronous data, which is provided for the CT1642 interface. The CT1642 chip samples LedData at the LedClk rising edge. After LedData is sampled for 18 clock cycles, LedClk is forced to high for a clock cycle. When the CT1642 chip samples a LedData rising edge, it latches the previously sampled 18 data segments, and then LedClk is forced to low for a clock cycle. When CT1642 chip samples a LedData rising edge again, it transmits the latched data.

The 18 data segments include four CSs, six useless signals, and eight LedData display data segments or KeyData scanning signals. The four CSs are valid alternatively. When a CS is valid, the transmitted LedData is the LED display data corresponding to the CS. The CSs become invalid after they are valid alternatively. In this case, no LED is selected, and the transmitted data is the key scanning signal corresponding to KeyData. The eight KeyData scanning signals are also valid alternatively.

Figure 11-26 Output timing of the LED module



11.5.3.2 Initialization

The LED module mainly provides two independent functions: keypad scanning and LED display control.

Before implementing keypad scanning, initialize the LED module as follows:

- Step 1** Configure clock dividers by writing the required clock dividers to the registers [LEDSYSTIM](#), [LEDCLKTIM](#), [LEDFRETIM](#), and [LEDKEYTIM](#).
- Step 2** Set the level when the LedCSx signal is selected, keypad scanning type, and the type of chip connected to the LED module by configuring [LEDCONFIG](#).
- Step 3** Enable keypad scanning by setting [LEDCONTROL](#) bit[3] to 1, select the interrupt mask mode by setting [LEDCONTROL](#) bit[1:0], and start the LED module by setting [LEDCONTROL](#) bit[9] to 1.



Step 4 Obtain keypad inputs by reading [LEDKEYDATA](#), and read the keypad interrupt information by reading [LEDKEYINT](#).

----End

Before controlling the LED display, initialize the LED module as follows:

Step 1 Configure the data to be displayed by writing the data to the registers [LEDDATA1](#), [LEDDATA2](#), [LEDDATA3](#), and [LEDDATA4](#).

Step 2 Configure clock dividers by writing the required clock dividers to the registers [LEDSYSTEM](#), [LEDCLKTIM](#), [LEDFRETIM](#), and [LEDFLASHTIM](#).

Step 3 Set the LED type, level of the LedCSx signal when the LED is on, and the type of the chip connected to the LED module by configuring [LEDCONFIG](#).

Step 4 Control the blinking enable bit corresponding to each LED by configuring [LEDCONTROL](#) bit[7:4], enable LED display by setting [LEDCONTROL](#) bit[8] to 1, and start the LED module by setting [LEDCONTROL](#) bit[9] to 1.

----End

11.5.4 Interface Signals

[Table 11-9](#) describes the LED interface signals.

Table 11-9 LED interface signals (8 x 1 matrix keypad scanning)

Signal	Width	Direction	Function Description
led_key0	1	I	Keypad input 0
led_clk	1	O	LED serial output data sync clock
led_data	1	O	LED serial output data

11.5.5 Register Summary

[Table 11-10](#) describes LED registers.

Table 11-10 Summary of LED registers (base address: 0xF800_3000)

Offset Address	Register	Description	Page
0x00	LEDCONTROL	LED control register	11-67
0x04	LEDCONFIG	LED display configuration register	11-69
0x08	LEDKEYINT	Keypad sampling interrupt register	11-70
0x0C	LEDKEYDATA	LED sampling keypad status register	11-71



Offset Address	Register	Description	Page
0x10	LEDCLKTIM	LedClk signal high/low level time parameter register	11-72
0x14	LEDFRETIM	LED refresh frequency register	11-72
0x18	LEDFLASHTIM	LED blinking frequency register	11-73
0x1C	LEDKEYTIM	Keypad scanning frequency register	11-73
0x20	LEDDATA1	First LED display data register	11-74
0x24	LEDDATA2	Second LED display data register	11-74
0x28	LEDDATA3	Third LED display data register	11-75
0x2C	LEDDATA4	Fourth LED display data register	11-75
0x34	LEDSYSTIM	LED system time frequency divider register	11-75

11.5.6 Register Description

LEDCONTROL

LEDCONTROL is a 10-bit read/write register. It is used to control LED display, blinking enable, keypad scanning enable, and keypad interrupts.



CAUTION

When the LED module works in CT1642 mode, you must disable the LED display by setting LEDCONTROL[led_dis_en] to 0 before disabling the LED module by setting LEDCONTROL[led_en] to 0. Otherwise, there may be residual characters on the LED.



Offset Address								Register Name								Total Reset Value																
0x00				LEDCONTROL								0x0000_0000																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																led_en	led_dis_en	flash_en4	flash_en3	flash_en2	flash_en1	key_en	reserved	int_press_mask	int_release_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			
Bits	Access	Name		Description																												
[31:10]	-	reserved		Reserved																												
[9]	RW	led_en		LED module enable 0: disabled 1: enabled																												
[8]	RW	led_dis_en		LED display enable 0: disabled 1: enabled																												
[7]	RW	flash_en4		Blinking enable for the fourth LED 0: disabled 1: enabled																												
[6]	RW	flash_en3		Blinking enable for the third LED 0: disabled 1: enabled																												
[5]	RW	flash_en2		Blinking enable for the second LED 0: disabled 1: enabled																												
[4]	RW	flash_en1		Blinking enable for the first LED 0: disabled 1: enabled																												
[3]	RW	key_en		Keypad enable 0: disabled 1: enabled																												
[2]	-	reserved		Reserved																												
[1]	RW	int_press_mask		Key press interrupt mask 0: masked 1: not masked																												



[0]	RW	int_release_mask	Key release interrupt mask 0: masked 1: not masked
-----	----	------------------	--

LEDCONFIG

LEDCONFIG is a 7-bit read/write register. It is used to set the type of the chip connected to the LED module, CS level when the LED is on, LED type, keypad scanning mode, and level of the keypad scanning CS.

	Offset Address	Register Name	Total Reset Value											
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	LEDCONFIG	0x0000_0000											
Name	reserved							leddata_dly_en	ledc_type	key8x1_cs	led_num	key_scan_mode	led_ty_cs	led_cs
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 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	leddata_dly_en	CT1642 output data delay enable 0: disabled 1: enabled											
[5]	RW	ledc_type	Type of the chip connected to the LED module. This bit must be set to 1 because only the CT1642 chip is supported. 0: reserved 1: CT1642											
[4]	RW	key8x1_cs	Level of the 8 x 1 matrix keypad scanning CS 0: low level 1: high level											
[3]	RW	led_num	Number of LED CSs. The CT1642 chip supports only four LED CSs. 0: four LED CSs 1: reserved											



[2]	RW	key_scan_mode	Keypad scanning mode. The CT1642 chip supports only the 8 x 1 scanning mode. 0: reserved 1: 8 x 1 scanning mode
[1]	RW	led_ty_cs	LED type 0: common-cathode LED 1: common-anode LED
[0]	RW	led_cs	LedCSx level when the LED is on. The level must be selected based on the board. 0: low level 1: high level

LEDKEYINT

LEDKEYINT is a 2-bit read/write register. It is a keypad sampling interrupt status register.

	Offset Address																Register Name								Total Reset Value											
	0x08																LEDKEYINT								0x0000_0000											
Bit	31	30	29	28	27	26	25	24	23	22	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_press		int_release						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	int_press		Key press interrupt flag 0: No interrupt is generated. 1: An interrupt is generated. Note: When a key release interrupt is generated, writing 1 to this bit clears the interrupt.																																
[0]	RW	int_release		Key release interrupt flag 0: No interrupt is generated. 1: An interrupt is generated. Note: When a key release interrupt is generated, writing 1 to this bit clears the interrupt.																																



LEDKEYDATA

LEDKEYDATA is an 8-bit read-only register. It is used to store the LED keypad sampling status.

In 8 x 1 scanning mode:

- For key 0, LedKey0 and CT1642 data signal 0 are valid.
- For key 1, LedKey0 and CT1642 data signal 1 are valid.
- For key 2, LedKey0 and CT1642 data signal 2 are valid.
- For key 3, LedKey0 and CT1642 data signal 3 are valid.
- For key 4, LedKey0 and CT1642 data signal 4 are valid.
- For key 5, LedKey0 and CT1642 data signal 5 are valid.
- For key 6, LedKey0 and CT1642 data signal 6 are valid.
- For key 7, LedKey0 and CT1642 data signal 7 are valid.

	Offset Address 0x0C																Register Name LEDKEYDATA								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								status_key7	status_key6	status_key5	status_key4	status_key3	status_key2	status_key1	status_key0
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RO	status_key7		Status of key 7 0: released 1: pressed																												
[6]	RO	status_key6		Status of key 6 0: released 1: pressed																												
[5]	RO	status_key5		Status of key 5 0: released 1: pressed																												
[4]	RO	status_key4		Status of key 4 0: released 1: pressed																												
[3]	RO	status_key3		Status of key 3 0: released 1: pressed																												



[2]	RO	status_key2	Status of key 2 0: released 1: pressed
[1]	RO	status_key1	Status of key 1 0: released 1: pressed
[0]	RO	status_key0	Status of key 0 0: released 1: pressed

LEDCLKTIM

LEDCLKTIM is a 4-bit read/write register. It is used to define the frequency of the LED serial sync clock LedClk.

Assume that the system clock is 24 MHz, the value of LEDClkTim is N, the frequency of the LED serial clock LedClk is F, and the LEDSysTim system time frequency divider is Q:

$$F = 24 / [(N + 1) \times 2 \times (Q + 1)] \text{ (unit: MHz)}$$

Offset Address								Register Name								Total Reset Value																
0x10				LEDCLKTIM								0x0000_0000																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								led_clk_tim							
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved			Reserved																											
[3:0]	RW	led_clk_tim			Frequency of the LED serial clock LedClk																											

LEDFRETIM

LEDFRETIM is a 4-bit read/write register. It is used to define the LED refresh frequency and the keypad scanning frequency in 4 x 2 or 8 x 1 scanning mode.

Assume that the system clock frequency is 24 MHz, the value of LEDClkTim is N, the value of LEDFreTim is M, the LED refresh frequency is F, the value of LEDSysTim is Q, and the number of LEDs is O.

The refresh frequency of the CT1642 LED is calculated as follows:

$$F = 24 / [(N + 1) \times (O + 1) \times 16 \times 40 \times (M + 1) \times (P + 1)] \text{ (unit: MHz)}$$



LEDFLASHTIM

LEDFLASHTIM is an 8-bit read/write register. It is used to define the LED blinking frequency.

Assume that the system clock frequency is 24 MHz, the value of LEDClkTim is N, the value of LEDFreTim is M, the value of LEDFlashTim is L, the LED blinking frequency is F, the value of LEDSysTim is Q, and the number of LEDs is O.

The refresh frequency of the CT1642 LED is calculated as follows:

$$F = 24 / [(N + 1) \times (Q + 1) \times 16 \times 40 \times (M + 1) \times (O + 1)] \text{ (unit: MHz)}$$

Offset Address								Register Name								Total Reset Value																
0x18								LEDFLASHTIM								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																led_flash_tim															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	led_flash_tim			LED blink frequency																											

LEDKEYTIM

LEDKEYTIM is a 4-bit read/write register. It is used to define the keypad scanning frequency.

Assume that the system clock frequency is 24 MHz, the value of LEDClkTim is N, the value of LEDFreTim is M, the value of LEDKeyTim is P, the keypad scanning frequency is F, the value of LEDSysTim is Q, and the number of LEDs is O.

The scanning frequency of the CT1642 keypad is calculated as follows:

$$F = 24 / [(N + 1) \times (Q + 1) \times 16 \times 40 \times (M + 1) \times (O + 1) \times 8 \times (P + 1)] \text{ (unit: MHz)}$$



Offset Address								Register Name								Total Reset Value																
0x1C				LEDKEYTIM								0x0000_0000																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								led_key_tim							
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved			Reserved																											
[3:0]	RW	led_key_tim			Keypad scanning frequency																											

LEDDATA1

LEDDATA1 is an 8-bit read/write register. It is used to configure the data displayed on the first LED.

Offset Address								Register Name								Total Reset Value																
0x20				LEDDATA1								0x0000_0000																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																led_data1															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	led_data1	Data displayed on the first LED																													

LEDDATA2

LEDDATA2 is an 8-bit read/write register. It is used to configure the data displayed on the second LED.

Offset Address								Register Name								Total Reset Value																
0x24				LEDDATA2								0x0000_0000																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																led_data2															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	led_data2			Data displayed on the second LED																											



LEDDATA3

LEDDATA3 is an 8-bit read/write register. It is used to configure the data displayed on the third LED.

Offset Address								Register Name								Total Reset Value																
0x28								LEDDATA3								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																led_data3															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	led_data3			Data displayed on the third LED																											

LEDDATA4

LEDDATA4 is an 8-bit read/write register. It is used to configure the data displayed on the fourth LED.

Offset Address								Register Name								Total Reset Value																
0x2C				LEDDATA4								0x0000_0000																				
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																led_data4															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	led_data4			Data displayed on the fourth LED																											

LEDSYSTIM

LEDSYSTIM is an LED system time frequency divider register. It is used to configure the global frequency divider for the LED clock.



Offset Address																Register Name								Total Reset Value								
0x34																LEDSYSTIM								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																								led_sys_tim							
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved	Reserved																													
[3:0]	RW	led_sys_tim	LED system time frequency divider																													

11.6 SPI

11.6.1 Overview

The SPI controller implements serial-to-parallel conversion and parallel-to-serial conversion, and can serve as a master to communicate with peripherals in sync serial mode. The SPI controller supports three peripheral interfaces including the SPI, TI serial sync interface, and microwire interface.

11.6.2 Features

The Hi3796M V100 SPI is a master interface and its working reference clock is 100 MHz. The SPI can output SPI_CLK with the maximum frequency of 50 MHz.

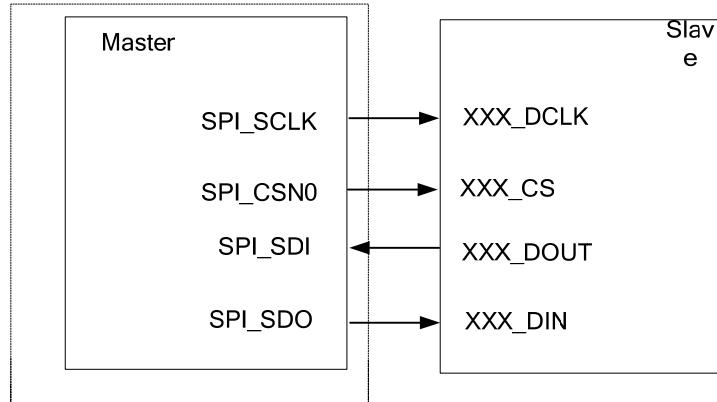
The SPI has the following features:

- Programmable interface clock frequency
- Two separate FIFOs, one acting as an RX FIFO and the other as a TX FIFO. Each FIFO is 16-bit wide and 256-location deep.
- 4-bit to 16-bit serial data frame
- Internal loopback test mode
- Three types of peripheral interfaces including the SPI, microwire interface, and TI sync serial interface
- Configurable full-duplex mode, clock polarity, and phase for the SPI
- Half-duplex mode for the microwire interface
- Full-duplex mode for the TI serial sync interface

11.6.3 Function Description

Typical Applications

Figure 11-27 shows the block diagram of the SPI connected to a single slave. The default CS pin SPI_CS0 is used.

Figure 11-27 SPI connected to a single slave

For details about how to select SPI CSs, see section 3.5 "Peripheral Controllers".

11.6.4 Peripheral Bus Timings

The abbreviations and acronyms in [Figure 11-28](#) to [Figure 11-35](#) are described as follows:

- MSB: most significant bit
- LSB: least significant bit
- Q: undefined signal

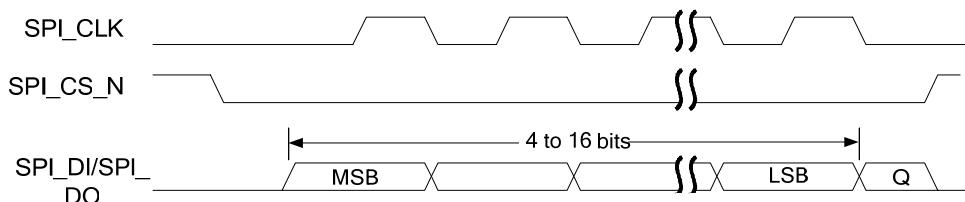
SPI Pins



SPO indicates the SPICLKOUT polarity, and SPH indicates the SPICLKOUT phase. The corresponding register bits are [SPICR0](#) bit[7:6].

(1) SPO = 0 and SPH = 0

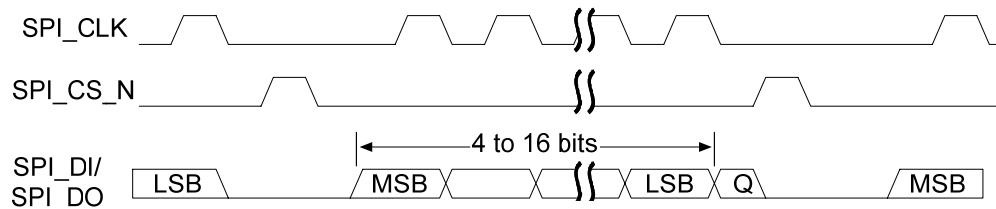
[Figure 11-28](#) shows the format of a single SPI frame.

Figure 11-28 Format of a single SPI frame (SPO = 0, SPH = 0)

[Figure 11-29](#) shows the format of consecutive SPI frames.



Figure 11-29 Format of consecutive SPI frames (SPO = 0, SPH = 0)



When the SPI is idle in this mode:

- The SPI_CLK signal is set to low.
- The SPI_CS_N signal is set to high.
- The TX data line SPI_DO is forced to low.

When the SPI is enabled and there is valid data in the TX FIFO, setting the SPI_CS_N signal to low starts a data transfer. The data from the slave device is immediately transmitted to the data RX line SPI_DI of the master device. Half an SPI_CLK cycle later, the valid master data is transmitted to SPI_DO. At this time, both the master data and slave data are valid. The SPI_CLK pin changes to high level half an SPI_CLK cycle later. Then, data is captured at the rising edge of the SPI_CLK clock and transmitted at the falling edge.

If a single word is transferred, SPI_CS_N is restored to high level one SPI_CLK cycle later after the last bit is captured.

For continuous transfer, the SPI_CS_N signal must be pulled up by one SPI_CLK cycle at each word transfer interval. This is because when SPH is 0, the slave select pin retains the data in the internal serial device register. Therefore, the master must pull the SPI_CS_N signal up at each word transfer interval in consecutive transfer. When the consecutive transfer ends, SPI_CS_N is restored to high one SPI_CLK cycle later after the last 1-bit data is captured.

(2) SPO = 0 and SPH = 1

Figure 11-30 shows the format of a single SPI frame.

Figure 11-30 Format of a single SPI frame (SPO = 0, SPH = 1)

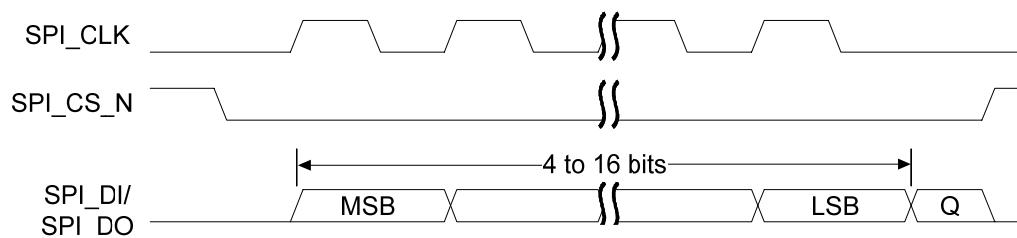
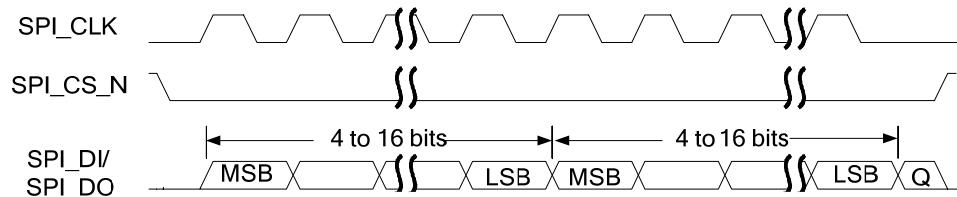


Figure 11-31 shows the format of consecutive SPI frames.



Figure 11-31 Format of consecutive SPI frames (SPO = 0, SPH = 1)



When the SPI is idle in this mode:

- The SPI_CLK signal is set to low.
- The SPI_CS_N signal is set to high.
- The TX data line SPI_DO is forced to low.

When the SPI is enabled and there is valid data in the TX FIFO, setting the SPI_CS_N signal to low starts a data transfer. If data transfer starts, the master data and slave data are valid on their respective transmission lines half an SPI_CLK cycle later. SPI_CLK becomes valid at the first rising edge. Data is captured at the falling edge of SPI_CLK and transmitted at the rising edge.

If a single word is transferred, SPI_CS_N is restored to high level one SPI_CLK cycle later after the data of the last bit is captured.

If consecutive words are transferred, SPI_CS_N retains low at the word transfer interval. When the consecutive transfer ends, SPI_CS_N is restored to high level one SPI_CLK cycle later after the last 1-bit data is captured.

(3) SPO = 1 and SPH = 0

Figure 11-32 shows the format of a single SPI frame.

Figure 11-32 Format of a single SPI frame (SPO = 1, SPH = 0)

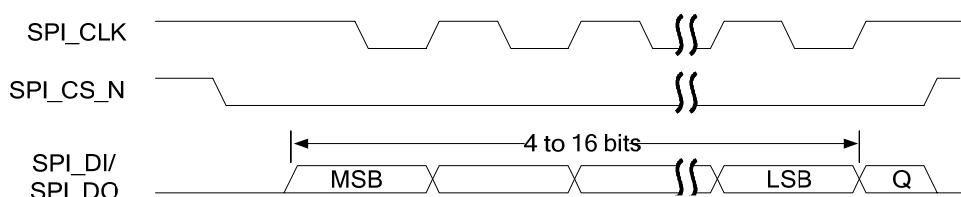
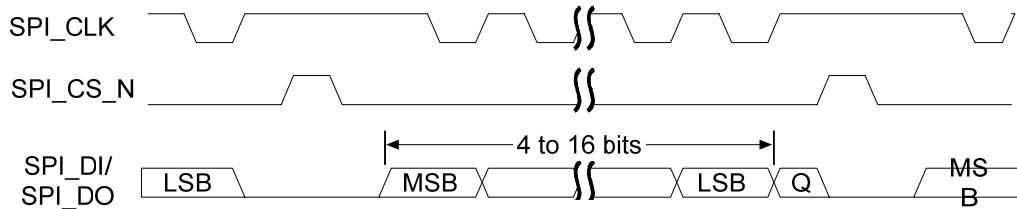


Figure 11-33 shows the format of consecutive SPI frames.



Figure 11-33 Format of consecutive SPI frames (SPO = 1, SPH = 0)



When the SPI is idle in this mode:

- The SPI_CLK signal is set to high.
- The SPI_CS_N signal is set to high.
- The TX data line SPI_DO is forced to low.

When the SPI is enabled and there is valid data in the TX FIFO, setting the SPI_CS_N signal to low starts a data transfer. The slave data is immediately transmitted to the master RX data line SPI_DI. Half an SPI_CLK cycle later, the valid master data is transmitted to SPI_DO. Another half SPI_CLK cycle later, the SPI_CLK master pin is set to low, indicating that data is captured at the falling edge of the SPI_CLK clock and transmitted at the rising edge.

If a single word is transferred, SPI_CS_N is restored to high level one SPI_CLK cycle later after the data of the last bit is captured.

If consecutive words are transferred, the SPI_CS_N signal must be pulled up at each word transfer interval. The reason is that when SPH is 0, the slave select pin fixes the data in the internal serial device register to maintain the data unchanged. SPI_CS_N is restored to high one SPI_CLK clock cycle later after the last bit is captured.

(4) SPH = 1 and SPO = 1

Figure 11-34 shows the format of a single SPI frame.

Figure 11-34 SPI single frame format (SPO = 1, SPH = 1)

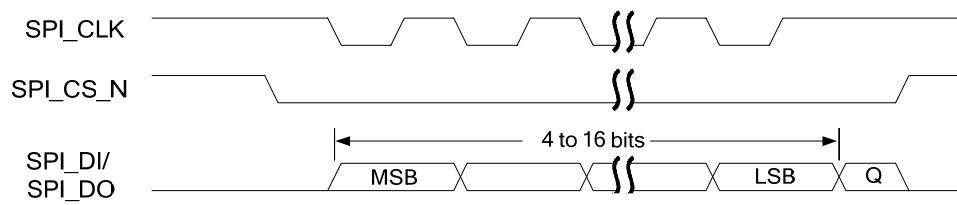
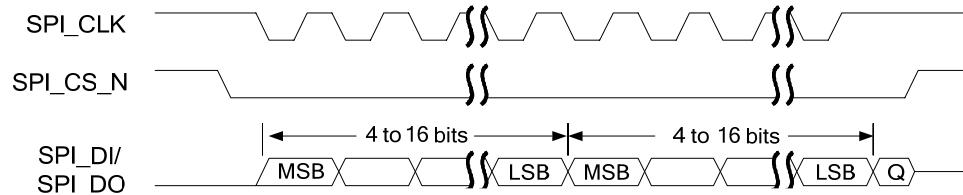


Figure 11-35 shows the format of consecutive SPI frames.



Figure 11-35 SPI consecutive frame format (SPO = 1, SPH = 1)



When the SPI is idle in this mode:

- The SPI_CLK signal is set to high.
- The SPI_CS_N signal is set to high.
- The TX data line SPI_DO is forced to low.

When the SPI is enabled and valid data is ready in the TX FIFO, setting the SPI_CS_N master signal to low starts the data transfer. Half an SPI_CLK cycle later, the master data and slave data are valid on their respective transmission lines. SPI_CLK becomes valid from a falling edge. Data is captured at the rising edge of the SPI_CLK clock and transmitted at the falling edge.

If a single word is transferred, SPI_CS_N is restored to high one SPI_CLK cycle later after the last bit is captured.

If consecutive words are transferred, the SPI_CS_N signal retains low. SPI_CS_N is restored to high one SPI_CLK cycle later after the last 1-bit data is captured. For consecutive transfer, SPI_CS_N retains low during data transfer, and the end mode is the same as that during single word transfer.

TI Serial Synchronous Interface

Figure 11-36 shows the format of a single frame for the TI synchronous serial interface.

Figure 11-36 Format of a single frame for the TI synchronous serial interface

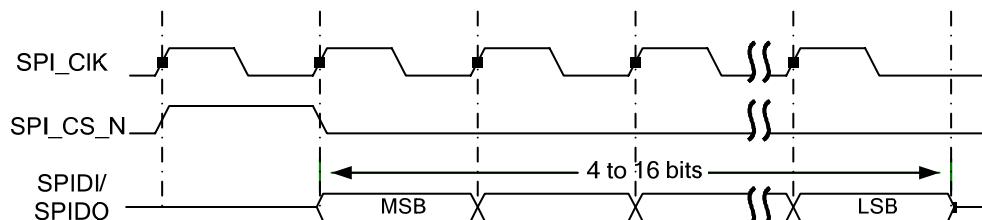
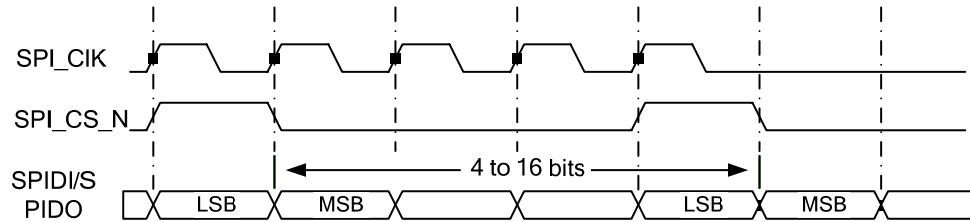


Figure 11-37 shows the format of consecutive frames for the TI synchronous serial interface.

Figure 11-37 Format of consecutive frames for the TI synchronous serial interface

When the SPI is idle in this mode:

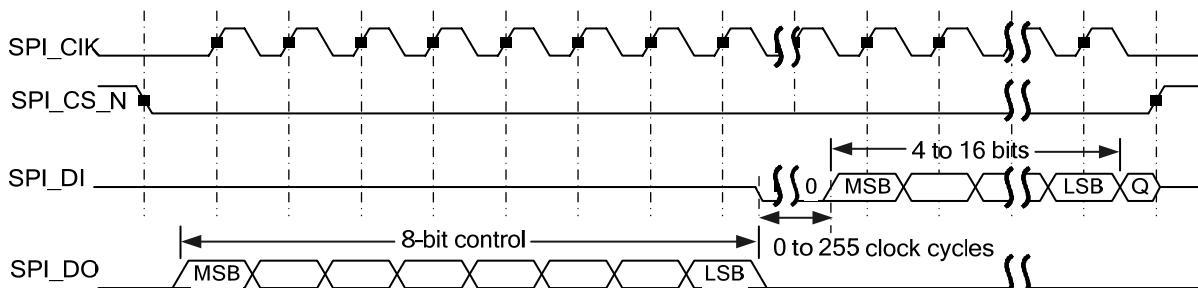
- The SPICK signal is set to low.
- The SPICSN signal is set to low.
- The TX data line SPIIDO retains high impedance.

If there is data in the TX FIFO, SPI_CS_N generates a high level pulse in one SPI_CK clock cycle. Then the data to be transmitted is transferred from the TX FIFO to the TX logic serial shift register. The MSBs of 4-bit to 16-bit data frames are shifted and output from SPI_DO at the next rising edge of the SPI_CK clock. Similarly, the MSBs of data received from the external serial slave device are shifted and input from the SPI_DI pin.

The SPI and off-chip serial device stores the data in the serial shift register at the falling edge of the SPI_CK clock. The RX serial register transmits the data to the RX FIFO at the rising edge of the first SPI_CK clock after receiving the LSB.

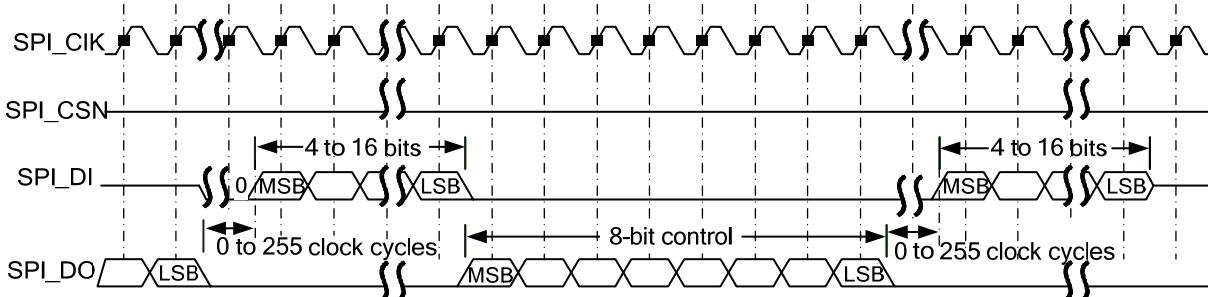
National Semiconductor Microwire Interface

[Figure 11-38](#) shows the format of a single frame for the national semiconductor microwire interface.

Figure 11-38 Format of a single frame for the national semiconductor microwire interface

0 to 255 clock cycles can be delayed between the end of the SPI_DO LSB and the start of the SPI_DI MSB.

[Figure 11-39](#) shows the format of consecutive frames for the national semiconductor microwire interface.

Figure 11-39 Format of consecutive frames for the national semiconductor microwire interface

0 to 255 clock cycles can be delayed between the end of the SPI_DO LSB and the start of the SPI_DI MSB.

The formats of the frames for the microwire interface and SPI are similar. Both of them use the master-slave transfer technology. The only difference is that the SPI works in full-duplex mode while the microwire interface works in half-duplex mode. Before serial data is transmitted to an external chip over the SPI, 8-bit control words are added. The SPI does not receive any data during this process. After the transfer is complete, the external chip decodes the received data. One clock cycle later after the 8-bit control information, the slave starts to acknowledge the required data. The returned data length is 4 bits to 16 bits, and therefore the length of an entire frame is 13 bits to 25 bits.

When the SPI is idle in this mode:

- The SPI_CLK signal is set to low.
- The SPI_CS_N signal is set to high.
- The TX data signal SPI_DO is forced to low level.

Writing one control byte to the TX FIFO starts a data transfer. The data transfer is triggered at the falling edge of SPI_CS_N. Data in the TX FIFO is transmitted to the serial shift register. The MSB of the 8-bit control frame is transmitted to the TX pin SPI_DO. During frame transfer, SPI_CS_N retains low, and SPI_DI retains high impedance.

The off-chip serial slave latches the data in the serial shift register at each rising edge of the SPI_CLK clock. When the slave latches the last 1-bit data, it starts to decode the received data after waiting one clock cycle, and then provides the required data to the SPI. Each bit is written to SPI_DI at the falling edge of the SPI_CLK clock. For a single data transfer, SPI_CS_N is pulled up at the end of the frame one clock cycle later after the last 1-bit data is written to the RX serial register, ensuring that the received data is transmitted to the RX FIFO.

The start and end of consecutive data transfer are the same as those of the single data transfer. During consecutive data transfer, the SPI_CS_N signal retains low, and the transferred data is consecutive. 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 fetched from the RX shift register at the falling edge of the SPI_CLK clock.

Interface Timings

Figure 11-40 SPI timing

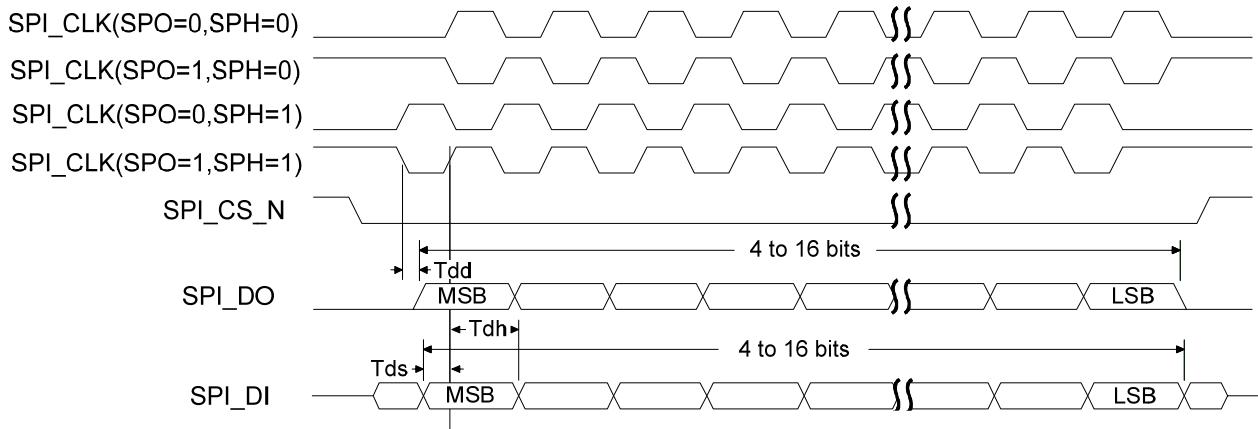


Table 11-11 SPI timing parameters

Parameter	Description	Min	Max	Unit
Tdd	Output data delay	-3.5	5	ns
Tds	Input control signal setup time	23	-	ns
Tdh	Input control signal hold time	0	-	ns

11.6.5 Operating Mode

The SPI supports data transfer in interrupt mode or query mode.

Clock and Reset

The frequency of the output SPI clock is calculated as follows:

$$F_{\text{SSPCLKOUT}} = F_{\text{SSPCLK}} / (\text{CPSDVR} \times (1 + \text{SCR}))$$

F_{SSPCLK} indicates the frequency of the SPI working reference clock, which is 100 MHz.

For details about CPSDVR and SCR, see the related registers.

The SPI working reference clock supports clock gating. The clock gating of SSP0 is controlled by configuring PERI_CRG28 bit[0]. Writing 0 to the corresponding bit disables the clock, and writing 1 to the corresponding bit enables the clock. The default value is 1 during power-on.

The SPI supports separate soft reset, which is controlled by configuring PERI_CRG28 bit[1]. Writing 0 to PERI_CRG28 bit[1] deasserts the soft reset on the SPI, and writing 1 to PERI_CRG28 bit[1] soft-resets the SPI. The default value is 0 during power-on.



Handling Interrupts

The SPI has the following five interrupts. The first four interrupts have independent interrupt sources and are maskable and active high.

- **SPIRXINTR**
RX FIFO interrupt. When there are four or more valid data segments in the RX FIFO, the interrupt is enabled.
- **SPITXINTR**
TX FIFO interrupt. When there are four or less valid data segments in the TX FIFO, the interrupt is enabled.
- **SPIRORINTR**
RX overflow interrupt. When the FIFO is full and new data needs to be written to the FIFO, FIFO overflow occurs and the interrupt is enabled. In this case, data is written to the RX shift register but not the FIFO.
- **SPIRTINTR**
RX timeout interrupt. When the RX FIFO is not empty and the SPI is idle for more than a fixed 32-bit cycle, the interrupt is enabled.
In this case, data in the RX FIFO needs to be transmitted. When the RX FIFO is read empty or new data is received in the SPIRXD, the interrupt is disabled. The interrupt can be cleared by writing to SPIICR[RTIC].
- **SPIINTR**
Combined interrupt, which is obtained after the preceding four interrupts are ORed. If any of the preceding four interrupts is enabled, this interrupt is enabled.

For details about SPIINTR, see section 3.6 "Interrupt System."

Initializing the SPI

The SPI is initialized as follows:

- Step 1** Write 0 to **SPICR1[sse]** to disable the SPI.
- Step 2** Write to **SPICR0** to set the parameters such as the frame format and transfer data bit width.
- Step 3** Configure **SPICPSR** to specify the required clock divider.
- Step 4** In interrupt mode, configure **SPIIMSC** to enable the corresponding interrupts; in query mode, disable the generation of interrupts.
- Step 5** In interrupt mode, configure **SPITXFIFOCCR** and **SPIRXFIFOCCR**.

----End

Transmitting Data in Query Mode

The depth of the TX or RX FIFO is 512 bytes.

Data is transmitted in query mode as follows:

- Step 1** If two CSs are used, select the CS to be operated by configuring PERI_CTRL bit[4].
- Step 2** Write 1 to **SPICR1[sse]** to enable the SPI.
- Step 3** Write the data to be transmitted to SPIDR continuously.



- Step 4** Poll **SPISR** until **SPISR[bsy]** is 0, indicating that the bus is not busy. If **SPISR[tfe]** is 1, the TX FIFO is empty; if **SPISR[rne]** is 1, the RX FIFO is not empty.
- Step 5** Read all data from the RX FIFO. You can check whether the RX FIFO is empty by querying **SPISR[rne]**.



CAUTION

As the SPI works in full-duplex mode, a data segment is received each time a data segment is transmitted. The RX FIFO must be cleared even data is only transmitted.

- Step 6** Write 0 to **SPICR1[sse]** to disable the SPI.

----End

Transmitting Data in Interrupt Mode

Data is transmitted in interrupt mode as follows:

- Step 1** If two CSs are used, select the CS to be operated by configuring PERI_CTRL bit[4].
- Step 2** Write 1 to **SPICR1[sse]** to enable the SPI.
- Step 3** Write the data to be transmitted to SPIDR continuously.
- Step 4** Wait for the SPIRXINTR interrupt to read data. Repeat this step until all data is read.



CAUTION

As the SPI works in full-duplex mode, a data segment is received each time a data segment is transmitted. The RX FIFO must be cleared even data is only transmitted.

- Step 5** Write 0 to **SPICR1[sse]** to disable the SPI.

----End

11.6.6 Register Summary

Table 11-12 describes SPI registers.

The base address for SPI registers is 0xF8B1_A000.

Table 11-12 Summary of SPI registers

Offset Address	Register	Description	Page
0x000	SPICR0	Control register 0	11-87
0x004	SPICR1	Control register 1	11-88



Offset Address	Register	Description	Page
0x008	SPIDR	Data register	11-89
0x00C	SPISR	Status register	11-90
0x010	SPICPSR	Clock divider register	11-90
0x014	SPIIMSC	Interrupt mask register	11-91
0x018	SPIRIS	Raw interrupt status register	11-91
0x01C	SPIMIS	Masked interrupt status register	11-92
0x020	SPIICR	Interrupt clear register	11-92
0x028	SPITXFIFOOCR	TX FIFO control register	11-93
0x02C	SPIRXFIFOOCR	RX FIFO control register	11-94

11.6.7 Register Description

SPICR0

SPICR0 is SPI control register 0.

Offset Address		Register Name										Total Reset Value		
Bit	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0	0x0000					
Name	scr								sph	spo	frf	dss		
Reset	0 0	0 0	0 0	0 0	0 0	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, ranging from 0 to 255 The field value is used to calculate the SPI TX and RX bit rates. The formula is as follows: Bit rate = $F_{SPICLK}/[CPSDVSR \times (1 + SCR)]$ CPSDVSR is an even number ranging from 2 to 254, and is specified by configuring SPICPSR.										
[7]	RW	sph		SPICLKOUT phase. For details, see section 11.6.4 "Peripheral Bus Timings."										
[6]	RW	spo		SPICLKOUT polarity. For details, see section 11.6.4 "Peripheral Bus Timings."										



[5:4]	RW	frf	Frame format 00: Motorola SPI frame 01: TI synchronous serial frame 10: national microwire frame 11: reserved
[3:0]	RW	dss	Data bit width 0011: 4 bits 1000: 9 bits 1101: 14 bits 0100: 5 bits 1001: 10 bits 1110: 15 bits 0101: 6 bits 1010: 11 bits 1111: 16 bits 0110: 7 bits 1011: 12 bits 0111: 8 bits 1100: 13 bits Other values: reserved

SPICR1

SPICR1 is SPI control register 1.

Offset Address 0x004												Register Name SPICR1				Total Reset Value 0x7F00			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	waiten	waitval												reserved	bigend	reserved	ms	sse	lmb
Reset	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name		Description															
[15]	RW	waiten		Wait enable. This bit is valid when SPICR0[frf] is set to the national microwire frame format. 0: disabled 1: enabled															



[14:8]	RW	waitval	Number of waiting beats between read and write operations in national microwire frame format. This field is valid when the waiten bit is 1 and the frame format is national microwire.
[7:5]	RW	reserved	Reserved
[4]	RW	bigend	Data endian mode 0: little endian 1: big endian
[3]	RW	reserved	Reserved
[2]	RW	ms	Master or slave mode. This field can be changed only when the SPI is disabled. 0: master mode (default) 1: reserved
[1]	RW	sse	SPI enable 0: disabled 1: enabled
[0]	RW	lbtm	Loopback mode 0: The normal serial port operation is enabled. 1: The output of the TX serial shift register internally connects to the input of the RX serial shift register.

SPIDR

SPIDR is a data register.

	Offset Address								Register Name								Total Reset Value							
	0x008								SPIDR								0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name																								
Reset																								
Bits	Access	Name		Description																				
[15:0]	RW	data		TX or RX FIFO Read: RX FIFO Write: TX FIFO If the number of data bits is less than 16, data must be right-aligned. The TX logic ignores the unused upper bits, and the RX logic automatically aligns the data to the right.																				



SPISR

SPISR is a status register.

	Offset Address								Register Name								Total Reset Value							
	0x00C								SPISR								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]	RW	reserved		Reserved																				
[4]	RW	bsy		SPI busy flag 0: ready 1: busy																				
[3]	RW	rff		Whether the RX FIFO is full 0: not full 1: full																				
[2]	RW	rne		Whether the RX FIFO is empty 0: empty 1: not empty																				
[1]	RW	tnf		Whether the TX FIFO is full 0: full 1: not full																				
[0]	RW	tfe		Whether the TX FIFO is empty 0: not empty 1: empty																				

SPICPSR

SPICPSR is a clock divider register.

	Offset Address								Register Name								Total Reset Value							
	0x010								SPICPSR								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]	RW	reserved		Reserved																				



[7:0]	RW	cpsdvsr	Clock divider The value must be an even number ranging from 2 to 254. It depends on the frequency of the input clock SPICLK. The LSB is read as 0.
-------	----	---------	---

SPIIMSC

SPIIMSC is an interrupt mask register. The value 0 indicates masked, and the value 1 indicates not masked.

	Offset Address		Register Name										Total Reset Value			
	0x014		SPIIMSC										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]	RW	reserved		Reserved												
[3]	RW	txim		TX FIFO interrupt mask 0: masked when the TX FIFO is half empty or less 1: not masked when the TX FIFO is half empty or less												
[2]	RW	rxim		RX FIFO interrupt mask 0: masked when the RX FIFO is half empty or less 1: not masked when the RX FIFO is half empty or less												
[1]	RW	rtim		RX timeout interrupt mask 0: masked 1: not masked												
[0]	RW	rorim		RX overflow interrupt mask 0: masked 1: not masked When the value is 1, the hardware stream control function is enabled. That is, when the RX FIFO is full, the SPI stops transmitting data.												

SPIRIS

SPIRIS is a raw interrupt status register. The value 0 indicates that no interrupt is generated, and the value 1 indicates that an interrupt is generated.



	Offset Address 0x018								Register Name SPIRIS								Total Reset Value 0x0008			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved														txris	rxris	rtris	rrorris		
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 TX FIFO interrupt status																
[2]	RO	rxris		Raw RX FIFO interrupt status																
[1]	RO	rtris		Raw RX timeout interrupt status																
[0]	RO	rrorris		Raw RX FIFO overflow interrupt status																

SPIMIS

SPIMIS is a masked interrupt status register. The value 0 indicates that no interrupt is generated, and the value 1 indicates that an interrupt is generated.

	Offset Address 0x01C								Register Name SPIMIS								Total Reset Value 0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved														txmis	rxmis	rtrmis	rrormis		
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		Masked TX FIFO interrupt status																
[2]	RO	rxmis		Masked RX FIFO interrupt status																
[1]	RO	rtrmis		Masked RX timeout interrupt status																
[0]	RO	rrormis		Masked RX FIFO overflow interrupt status																

SPIICR

SPIICR is an interrupt clear register. Writing 1 to the corresponding bit clear an interrupt, and writing 0 has no effect.



Offset Address										Register Name					Total Reset Value				
Bit	0x020					SPIICR					0x0000								
Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
reserved											rtic				roric				
Reset	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]	RO	rtic			RX timeout interrupt clear														
[0]	RO	roric			RX overflow interrupt clear														

SPITXFIFOOCR

SPITXFIFOOCR is a TX FIFO control register.

Offset Address										Register Name					Total Reset Value				
Bit	0x028					SPITXFIFOOCR					0x0009								
Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
reserved											txintsize				dmatxbysize				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1			
Bits	Access	Name			Description														
[15:6]	RW	reserved			Reserved														
[5:3]	RW	txintsize			Threshold for triggering a TX FIFO interrupt. When the amount of data in the TX FIFO is less than or equal to the value of TXINTSize, TXRIS is valid. 000: 1 001: 4 010: 8 011: 16 100: 32 101: 64 110: 64 111: 64														



[2:0]	RO	reserved	Reserved	
-------	----	----------	----------	--

SPIRXFIFOOCR

SPIRXFIFOOCR is an RX FIFO control register.

Offset Address 0x02C										Register Name SPIRXFIFOOCR				Total Reset Value 0x0009			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved										rxintsize					dmarsize	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
Bits	Access	Name		Description													
[15:6]	RW	reserved		Reserved													
[5:3]	RW	rxintsize		Threshold for triggering an RX FIFO interrupt. When the amount of data in the RX FIFO is greater than or equal to the value of (256 - RXINTSize), RXRIS is valid and its length is 16 bits. 000: 1 001: 4 010: 8 011: 16 100: 32 101: 64 110: 64 111: 64													
[2:0]	RO	reserved		Reserved													

11.7 USB 2.0

11.7.1 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, OHCI, and EHCI protocols. The USB 2.0 host controller has a root hub that is a part of the USB system and is used to extend the USB port. Most hardware logic of the USB 2.0 host controller supports the following functions:

- Controls and processes data transfer.
- Parses data packets and packages data.
- Encodes and decodes the signals transmitted over the USB port.
- Provides interfaces (such as the interrupt vector interface) for the driver.

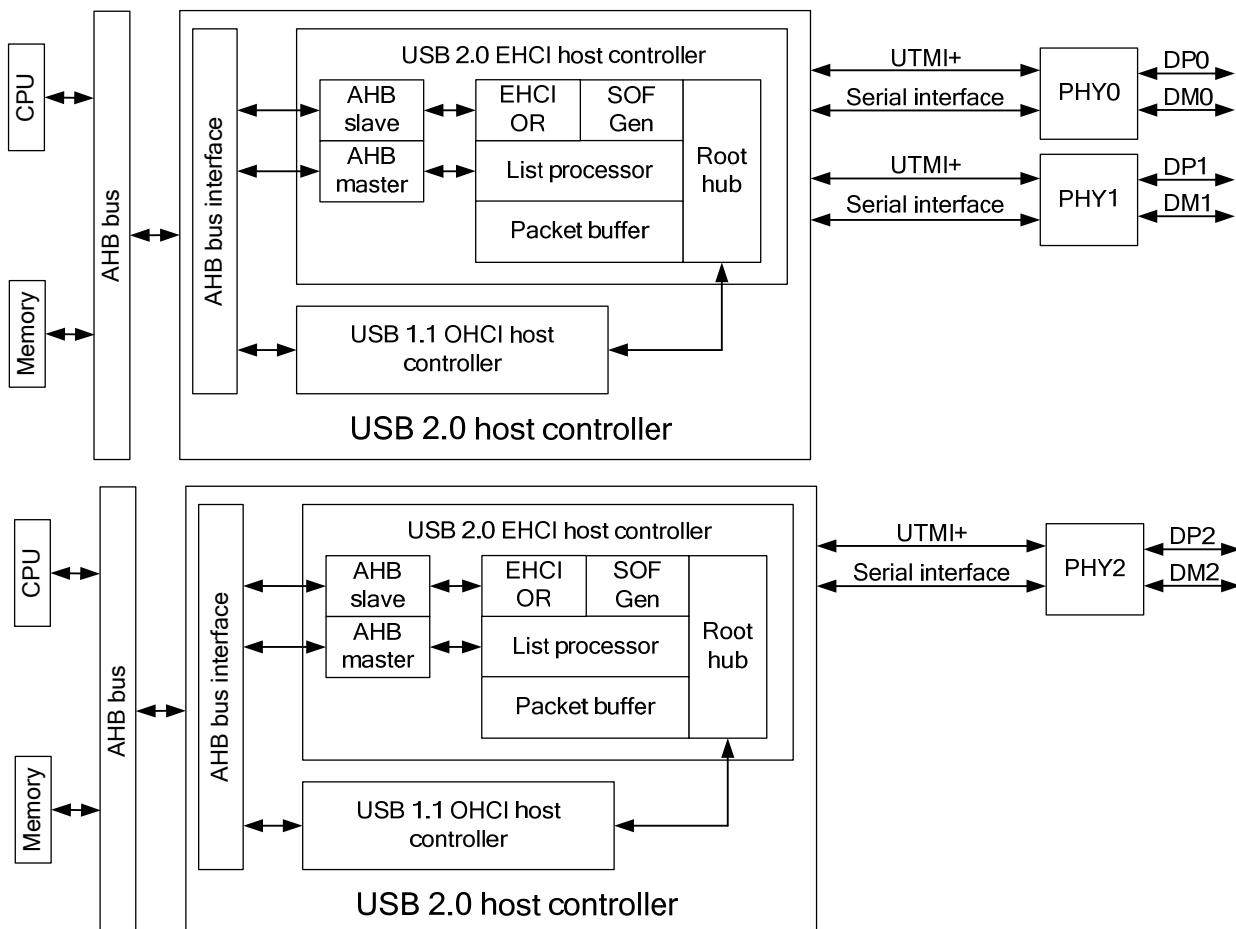
The USB 2.0 device controller supports the high-speed (480 Mbit/s) and full-speed (12 Mbit/s) data transfer modes. Port 0 allows intelligent switchover between the host and device.

11.7.2 Function Description

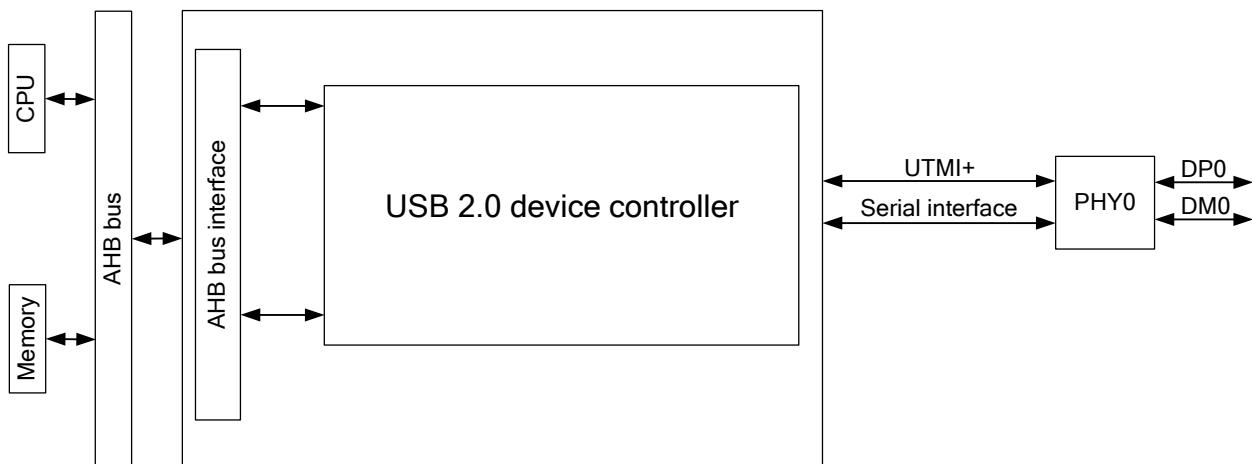
Logic Block Diagram

[Figure 11-41](#) shows the logic block diagram of the USB 2.0 host controller.

Figure 11-41 Logic block diagram of the USB 2.0 host controller



[Figure 11-42](#) shows the logic block diagram of the USB 2.0 device controller.

Figure 11-42 Logical block diagram of the USB 2.0 device controller

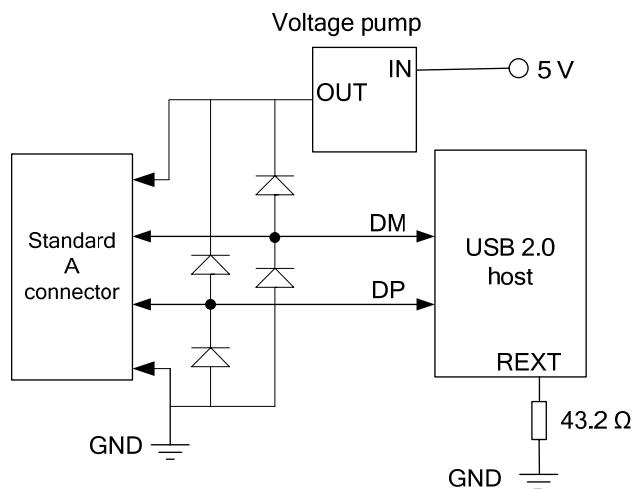
Typical Applications

Figure 11-43 shows the reference design of the USB 2.0 host controller.



CAUTION

- The single-ended impedance of DP or DM is $45\ \Omega \pm 1\%$, and no extra matched resistor is required.
- The precision of the REXT resistor is $\pm 1\%$.
- If high-speed electro static discharge (ESD) components are used, 1 pF capacitors are recommended.

Figure 11-43 Reference design of the USB 2.0 module



Features

The USB 2.0 host controller has the following features:

- Complies with the USB 2.0 standard.
- Complies with the OHCI and EHCI protocols.
- Supports high-speed, full-speed, and low-speed devices.
- Supports low-power solutions.
- Supports four basic data transfer modes: control transfer, bulk transfer, isochronous transfer, and interrupt transfer.
- Supports a maximum of 127 devices by using USB hubs.

The USB 2.0 device controller has the following features:

- Complies with the USB 2.0 standard.
- Supports the high-speed and full-speed modes.
- Supports two basic data transfer modes: control transfer and bulk transfer.

Function Implementation

The USB 2.0 module supports the following four standard transfer modes:

- Control transfer

This mode applies to the data transfer between endpoints 0 of the USB host and USB device. For the USB devices of specific models, other endpoints may be used. The control transfer is bidirectional and the transferred data amount is small. Depending on the device and transfer speed, 8-byte data, 16-byte data, 32-byte data, or 64-byte data can be transferred.

- Bulk transfer

This mode is typically used when a large amount of data is transmitted with no requirements on the bandwidth and time interval. This mode is the best choice when the transfer speed is very low and many data transfers are delayed. Bulk transfer is performed after all other types of data transfers are complete. This mode ensures that data is transferred between the USB host and USB device without errors by using an error detection and retransmission mechanism.

- Isochronous transfer

This mode applies to the stream data transfer with strict time requirements and high error tolerance or the instant data transfer at a constant transfer rate. This mode provides a specific bandwidth and time interval.

- Interrupt transfer

This mode applies to transfer of small-sized, scattered, and unpredictable data. In this mode, the device is regularly queried for interrupt data to be sent. The query frequency ranges from 1 ms to 255 ms and it depends on the device endpoint mode. Typically, the interrupt transfer is unidirectional and only input is available for the USB host.

11.7.3 Operating Mode

Controlling the Clock Gating

If the USB 2.0 host controller is not used, its clocks can be disabled to reduce power consumption.



The working clock is disabled as follows:

Step 1 Set the corresponding ports to suspend mode by using the drivers complying with the EHCI and OHCI protocols.

Step 2 Set PERI_CRG46 bit[6:0] and PERI_CRG102 bit[5:0] to 0.

----End

The working clock is enabled as follows:

Step 1 Set PERI_CRG46 bit[6:0] and PERI_CRG102 bit[5:0] to 1 to enable the clocks of the USB 2.0 host controller.

Step 2 Enable ports to exit the suspend mode by using the drivers complying with the EHCI and OHCI protocols.

----End

Deasserting Reset

The USB controller and PHY are reset after power-on by default. The reset is deasserted as follows:

Step 1 Delay for at least 10 μ s.

Step 2 Write 0 to PERI_CRG100 bit[8] and PERI_CRG47 bit[8] to deassert the global reset on the USB PHY.

Step 3 Wait 250 μ s until the internal PLL of the PHY is stable.

Step 4 Read PERI_CRG91 bit[2:0]. If the bits are all 1s, go to step 6; otherwise, go to step 5.

Step 5 Write 1 to PERI_CRG100 bit[8] and PERI_CRG47 bit[8] to reset the PHY, wait 10 μ s, and then go to step 2.

Step 6 Write 0 to PERI_CRG46[17:12] and PERI_CRG102[16:12] to deassert resets on each part.

----End

Separately Resetting Ports During the Working Process

Ports are separately reset during the working process as follows:

Step 1 Write 1 to PERI_CRG102 bit[5], PERI_CRG46 bit[14], or PERI_CRG46 bit[13] (corresponding to port 2, port 1, and port 0 respectively) to soft-reset the USB controller port.

Step 2 Write 1 to PERI_CRG100 bit[9], PERI_CRG47 bit[11], or PERI_CRG47 bit[9] (corresponding to port 2, port 1, and port 0 respectively) to soft-reset the USB PHY port.

Step 3 Wait 200 μ s, and then write 0 to PERI_CRG100 bit[9], PERI_CRG47 bit[11], or PERI_CRG47 bit[9] (corresponding to port 2, port 1, and port 0 respectively) to deassert the reset on the USB PHY port.

Step 4 Write 0 to PERI_CRG102 bit[5], PERI_CRG46 bit[14], or PERI_CRG46 bit[13] (corresponding to port 2, port 1, and port 0 respectively) to deassert the reset on the USB controller port.

----End



Suspending and Resuming a Port

Suspending a port: A port enters the suspend mode after software enables the suspend mode by configuring the EHCI/OHCI register.

Resuming a port: After software enables the port to exit the suspend mode by configuring the EHCI/OHCI register, it can initiate a USB operation only after at least 250 μ s delay.

11.7.4 Summary of USB Host Register

Table 11-13 describes USB host registers.

- Base address for port 0 and port 1 registers: 0xF989_0000
- Base address for port 2 registers: 0xF993_0000

Table 11-13 Summary of USB host registers

Offset Address	Register	Description	Page
0x90	INSNREG00	Micro-frame length configuration register	11-99
0x94	INSNREG01	PBUF out/in threshold configuration register	11-100
0x98	INSNREG02	PBUF depth configuration register	11-100
0x9C	INSNREG03	Interrupt memory transfer enable register	11-101
0xA0	INSNREG04	Debug register	11-101
0xA4	INSNREG05	Control and status register	11-102
0xA8	INSNREG06	AHB error status register	11-103
0xAC	INSNREG07	AHB error address register	11-104

11.7.5 Description of USB Host Registers

INSNREG00

INSNREG00 is a micro-frame length configuration register.

Bit	Offset Address																Register Name								Total Reset Value												
	0x90																INSNREG00								0x0000_0000												
Name	reserved																debug				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																															
[31:20]	-	reserved				Reserved																															



[19:14]	RW	debug	For debugging only
[13:1]	RW	val	Micro-frame counter This field is used only for simulation. In normal cases, the micro-frame length is 125 μs defined in the protocol. During simulation, the micro-frame length can be shortened by configuring this field to reduce the simulation time.
[0]	RW	en	Register enable 0: disabled 1: enabled

INSNREG01

INTNREG01 is a PBUF out/in threshold register.

	Offset Address 0x94																Register Name INSNREG01								Total Reset Value 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		
Bits	Access	Name				Description																											
[31:16]	RW	out_threshold				TX threshold If the data amount in the PBUF is above the TX threshold, data starts to be transmitted. The depth is measured by 32 bits.																											
[15:0]	RW	in_threshold				RX threshold If the data amount in the PBUF is above the RX threshold, data is read from the PBUF. The depth is measured by 32 bits.																											

INSNREG02

INSNREG02 is a PBUF depth configuration register.

	Offset Address 0x98																Register Name INSNREG02								Total Reset Value 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																pbuf_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	
Bits	Access	Name				Description																										
[31:12]	-	reserved				Reserved																										



[11:0]	RW	pbuf_depth	PBUF depth. The depth is measured by 32 bits.
--------	----	------------	---

INSNREG03

INSNREG03 is an interrupt memory transfer enable register.

	Offset Address 0x9C																Register Name INSNREG03								Total Reset Value 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																dis	ctrl	val	fetch	offset								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	1		
Bits	Access	Name	Description																													
[31:15]	-	reserved	Reserved																													
[14]	RW	dis	256 MHz clock check enable 0: disabled 1: enabled																													
[13]	RW	ctrl	Line state ignore during TESTSE0 NAK																													
[12:10]	RW	val	TX-TX turnaround delay attach																													
[9]	RW	fetch	Periodic frames list RX																													
[8:1]	RW	offset	Available time offset																													
[0]	RO	brk_en	Interrupt memory transfer enable 0: disabled 1: enabled																													

INSNREG04

INSNREG04 is a debug register.



	Offset Address	Register Name	Total Reset Value															
Bit	0xA0	INSNREG04	0x0000_0000															
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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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: enabled (default). The suspend signal is valid when the run/stop bit is reset by software, but the hchalted bit is not set. 1: disabled. The port is not suspended when software clears the run/stop bit.															
[4]	RW	nak_reldfix_en	NAK reload enable 0: enabled 1: disable															
[3]	-	reserved	Reserved															
[2]	RW	scaledwn_enum_time	Port enumeration time scale-down enable 0: disabled 1: enabled															
[1]	RW	hccparam_en	HCCPARAMS register write enable 0: disabled 1: enabled															
[0]	RW	hcsparam_en	HCSPARAMS register write enable 0: disabled 1: enabled															

INSNREG05

INSNREG05 is a control and status register. It is used to read or write to PHY registers.



Offset Address																Register Name								Total Reset Value								
0xA4																INSNREG05								0x4000_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		hccparam_en									
Reset	0	1	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	
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. It cannot exceed the supported number of ports.																												
[12]	RW	vcontrol_loadm		Load enable 0: enabled 1: disable																												
[11:8]	RW	vcontrol		Port control signal																												
[7:0]	RO	hccparam_en		Port status signal																												

INSNREG06

INSNREG06 is an AHB error status register.



INSNREG07

INSNREG07 is an AHB error address register.

Offset Address								Register Name								Total Reset Value																
0x0AC				INSNREG07								0x0000_0000																				
Bit	31	30	29	28	27	26	25	24	23	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		
Bits	Access	Name			Description																											
[31:0]	RO	err_addr			Address during control transfer when an AHB error occurs																											



11.7.6 Offset Address Variables for USB Device Registers

Table 11-14 describes the value range and meaning of the variables in the offset addresses for USB device registers.

Table 11-14 Register variables

Variable	Value Range	Description
FIFO_num	0–14	Number of FIFOs
n	0–15	Endpoint n

11.7.7 Summary of USB Device Registers

Table 11-15 describes USB device registers.

Table 11-15 Summary of USB device registers (base address: 0xF98C0000)

Offset Address	Register	Description	Page
0x0000	GOTGCTL	Device behavior control and status query register	11-107
0x0004	GOTGINT	Device interrupt generation indicator/clear register	11-110
0x0008	GAHBCFG	AHB configuration register	11-110
0x000C	GUSBCFG	USB configuration register	11-112
0x0010	GRSTCTL	Reset hardware feature register	11-115
0x0014	GINTSTS	System interrupt register	11-117
0x0018	GINTMSK	System interrupt mask register	11-120
0x001C	GRXSTSR	RX status debug read register	11-124
0x0020	GRXSTSP	RX status read&pop register	11-125
0x0024	GRXFSIZ	RX FIFO size configuration register	11-126
0x0028	GNPTXFSIZ	Non-periodic TX FIFO size configuration register	11-126
0x002C	GNPTXSTS	Non-periodic TX FIFO and non-periodic TX request queue register	11-126
0x0030	GI2CCTL	I ² C access register	11-127
0x0034	GPVNDCTL	PHY vendor control register	11-129
0x0038	GPIO	GPIO register	11-130
0x003C	GUID	User ID query register	11-130
0x0040	GSNPSID	Synopsys ID query register	11-130
0x0044	GHWCFG1	User hardware configuration register 1	11-131



Offset Address	Register	Description	Page
0x0048	GHWCFG2	User hardware configuration register 2	11-131
0x004C	GHWCFG3	User hardware configuration register 3	11-133
0x0050	GHWCFG4	User hardware configuration register 4	11-135
0x0054	GLPMCFG	Link power management (LPM) configuration register	11-137
0x0058	GPWRDN	Power-down register	11-139
0x005C	GDFIFO CFG	DFIFO software configuration register	11-141
0x0060	GADPCTL	ADP timer control and status register	11-142
0x0100	HPTXFSIZ	TX FIFO configuration register	11-144
0x0104+0x0004×F IFO_num	DPTXFSIZN	Device periodic TX FIFO-n size register	11-144
0x0104+0x0004×F IFO_num	DIEPTXFN	Device IN endpoint TX FIFO size register	11-145
0x0800	DCFG	Device configuration register	11-145
0x0804	DCTL	Device control register	11-147
0x0808	DSTS	Device status register	11-149
0x0810	DIEPMSK	Device IN endpoint common interrupt mask register	11-149
0x0814	DOEPMSK	Device OUT endpoint common interrupt mask register	11-151
0x0818	DAINT	Interrupt register for all device endpoints	11-152
0x081C	DAINTMSK	Interrupt mask register for all device endpoints	11-153
0x0820	DTKNQR1	Device IN token sequence learning queue read register 1	11-153
0x0824	DTKNQR2	Device IN token sequence learning queue read register 2	11-154
0x0830	DTKNQR3	Device IN token sequence learning queue read register 3	11-155
0x0834	DTKNQR4	Device IN token sequence learning queue read register 4	11-155
0x0828	DVBUSDIS	Device VBUS discharge time register	11-155
0x082C	DVBUSPULSE	Device VBUS pulsing time register	11-156
0x0830	DTHRCTL	Device threshold control register	11-156
0x0834	DIEPEMPMSK	Device IN endpoint FIFO empty interrupt mask register	11-157
0x0838	DEACHINT	Interrupt register for each device endpoint	11-158
0x083C	DEACHINTMSK	Interrupt mask register for each device endpoint	11-158
0x0840 + (0x0004×n)	DIEPEACHMSK_N	Interrupt register for device IN endpoint n	11-159



Offset Address	Register	Description	Page
0x0880 + (0x0004 x n)	DOEPEACHMS KN	Interrupt register for device OUT endpoint n	11-160
0x0900	DIEPCTL0	Control register for device control IN endpoint 0	11-162
0x0B00	DOEPCTL0	Control register for device control OUT endpoint 0	11-163
0x0900 + (0x0020 x n)	DIEPCTLN	Control register for device IN endpoint n	11-165
0x0B00 + (0x0020 x n)	DOEPCTLN	Control register for device OUT endpoint n	11-166
0x0908 + (0x0020 x n)	DIEPINTn	Interrupt register for device IN endpoint n	11-168
0x0B08 + (0x0020 x n)	DOEPINTn	Interrupt register for device OUT endpoint n	11-171
0x0910	DIEPTSIZ0	Transfer size register for device IN endpoint 0	11-173
0x0B10	DOEPTSIZ0	Transfer size register for device OUT endpoint 0	11-174
0x0910 + (0x0020 x n)	DIEPTSIZn	Transfer size register for device IN endpoint n	11-175
0x0B10 + (0x0020 x n)	DOEPTSIZn	Transfer size register for device OUT endpoint n	11-175
0x0914 + (0x0020 x n)	DIEPDMAN	DMA address register for device IN endpoint n	11-176
0x0B14 + (0x0020 x n)	DOEPDMAN	DMA address register for device OUT endpoint n	11-177
0x091C + (0x0020 x n)	DIEPDMABN	DMA buffer address register for device IN endpoint n	11-177
0x0B1C + (0x0020 x n)	DOEPDMABN	DMA buffer address register for device OUT endpoint n	11-178
0x0938	DTXFSTS n	Device IN endpoint TX FIFO status register	11-178

11.7.8 Description of USB Device Registers

GOTGCTL

GOTGCTL is a device behavior control and status query register.



	Offset Address 0x0000												Register Name GOTGCTL												Total Reset Value 0x04C1_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved				multvalidbc				reserved	otgver	bsesvld	asesvld	dbnctime	conidsts	reserved				devhnpnpen	hstsethnpnpen	hmpreq	hstneges	bvaliddovval	bvaliddovn	avaliddovn	avaliddovval	vbvaliddovn	vbvaliddovval	sesreq	sesreqscs						
Reset	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
Bits	Access	Name		Description																																
[31:27]	-	reserved		Reserved																																
[26:22]	RO	multvalidbc		BC ACA input Bit[26]: rid_float Bit[25]: rid_gnd Bit[24]: rid_a Bit[23]: rid_b Bit[22]: rid_c																																
[21]	RO	reserved		Reserved																																
[20]	RW	otgver		Device version 0: version 1.3 1: version 2.0																																
[19]	RO	bsesvld		Transceiver status in device mode 0: B_session is invalid. 1: B_session is valid.																																
[18]	RO	asesvld		Transceiver status in host mode 0: A_session is invalid. 1: A_session is valid.																																
[17]	RO	dbnctime		Dejitter time 0: long dejitter time 1: short dejitter time																																
[16]	RO	conidsts		USB_ID status 0: The device works in A-device mode. 1: The device works in B-device mode.																																
[15:12]	RO	reserved		Reserved																																
[11]	RW	devhnpnpen		Device Host Negotiation Protocol (HNP) enable 0: disabled 1: enabled																																



[10]	RW	hstsethnpn	Host HNP enable 0: disabled 1: enabled
[9]	RW	hnpreq	HNP request 0: no request 1: request
[8]	RO	hstnegscs	Host negotiation indicator 0: Host negotiation fails. 1: Host negotiation succeeds.
[7]	RW	bvalidovval	Bvalid setting 0: Bvalid = 0 1: Bvalid = 1 This bit is valid when GOTGCTL[bvalidoven] is 1.
[6]	RW	bvalidoven	Bvalid signal overwrite enable 0: Bvalid can be overwritten. 1: Bvalid cannot be overwritten.
[5]	RW	avalidovval	Avalid setting 0: Avalid = 0 1: Avalid = 1 This bit is valid when GOTGCTL[avalidoven] is 1.
[4]	RW	avalidoven	Avalid signal overwrite enable 0: Avalid can be overwritten. 1: Avalid cannot be overwritten.
[3]	RW	vbusvalidovval	vbusvalid setting 0: vbusvalid = 0 1: vbusvalid = 1 This bit is valid when GOTGCTL[vbusvalidoven] is 1.
[2]	RW	vbusvalidoven	vbusvalid signal overwrite enable 0: vbusvalid can be overwritten. 1: vbusvalid cannot be overwritten.
[1]	RW	sesreq	Session request 0: no request 1: request
[0]	RO	sesreqscs	Session request status 0: failed 1: succeeded



GOTGINT

GOTGINT is a device interrupt generation indicator/clear register.

	Offset Address										Register Name										Total Reset Value											
	0x0004										GOTGINT										0x0000_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved										dbncedone	adevtoutchg	hstnegdet	reserved						hstnegsucstschn	sesreqsucstschn	reserved				sesenddet	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		
Bits	Access	Name		Description																												
[31:20]	RO	reserved		Reserved																												
[19]	RO	dbncedone		An interrupt is generated when dejitter is successful. Setting this bit to 1 clears the interrupt. This bit is valid only when the HNP capable or SRP capable bit is set to 1.																												
[18]	RO	adevtoutchg		An interrupt is generated when waiting for the connection of device B times out. Setting this bit to 1 clears the interrupt.																												
[17]	RO	hstnegdet		An interrupt is generated when host negotiation is detected. Setting this bit to 1 clears the interrupt.																												
[16:10]	RO	reserved		Reserved																												
[9]	RO	hstnegsucstschn		An interrupt is generated when the host negotiation request fails or succeeds. Setting this bit to 1 clears the interrupt.																												
[8]	RO	sesreqsucstschn		An interrupt is generated when the session request fails or succeeds. Setting this bit to 1 clears the interrupt.																												
[7:3]	RO	reserved		Reserved																												
[2]	RO	sesenddet		An interrupt is generated when utmiotg_bvalid is deasserted. Setting this bit to 1 clears the interrupt.																												
[1:0]	RO	reserved		Reserved																												

GAHBCFG

GAHBCFG is an AHB configuration register.



	Offset Address	Register Name	Total Reset Value
Bit	0x0008	GAHBCFG	0x0000_0000
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	reserved	invdescendianness ahbsingle notialldmawrit remmemsupp reserved	nptxfemplvl reserved dmaen hbstlen glblintrmsk
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 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]	RW	reserved	Reserved
[24]	RW	invdescendianness	Descriptor byte invert 0: The descriptor byte sequence is similar to the AHB master byte sequence. 1: When the AHB master byte sequence is big endian, the descriptor byte sequence is little endian, and versa vice.
[23]	RW	ahbsingle	Transfer in DMA mode 0: The remaining data is transmitted based on the INCR burst size. 1: The remaining data is transmitted based on the single burst size.
[22]	RW	notialldmawrit	DMA write operation notification. This bit is valid only when GAHBCFG[remmemsupp] is 1.
[21]	RW	remmemsupp	Remote memory support
[20:8]		reserved	Reserved
[7]	RW	nptxfemplvl	Empty status level of the non-periodic TX FIFO
[6]		reserved	Reserved
[5]	RW	dmaen	DMA mode enable 0: The core works in slave mode. 1: The core works in DMA mode.



[4:1]	RW	hbstlen	Burst length/type for both the external and internal DMA modes In external DMA mode: 0000: 1 word 0001: 4 words 0010: 8 words 0011: 16 words 0100: 32 words 0101: 64 words 0110: 128 words 0111: 256 words Other values: reserved In internal DMA mode: 0000: Single 0001: INCR 0011: INCR4 0101: INCR8 0111: INCR16 Other values: reserved
[0]	RW	glblintrmsk	Global interrupt mask 0: masked 1: not masked

GUSBCFG

GUSBCFG is a USB configuration register.

Bit	Offset Address 0x000C																Register Name GUSBCFG								Total Reset Value 0x0000_1400									
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	corrupttxpkt	forcedemode	forcehstmode	txnddelay	reserved	ic_usbcap	ulp <i>i</i>	indicator	complement	termeldpulse	reserved	ulpifsls	reserved	phylpwrclksel	reserved	usbtrdtim	lmpcap	srgcap	reserved	physel	fsintf	ulp <i>i</i> _utmi_sel	phyif	toutcal										
Reset	0	0	0	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			
Bits	Access	Name				Description																												
[31]	WO	corrupttxpkt				Unexpected TX packet This bit is for debugging only and is always set to 0.																												



[30]	RW	forcedevmode	Forcible device mode 0: common mode 1: forcible device mode
[29]	RW	forcehstmode	Forcible host mode 0: common mode 1: forcible host mode
[28]	RW	txenddelay	TX end delay enable 0: disabled 1: enabled
[27]	-	reserved	Reserved
[26]	RO	ic_usbcap	IC_USB enable 0: The IC_USB PHY interface is not selected. 1: The IC_USB PHY interface is selected.
[25]	RW	ulpri	UTMI+low pin interface (ULPI) protection enable 0: enabled 1: disabled
[24]	RO	indicator	Indicator pass-through 0: The complementary output signal is determined by the vbusvalid comparator. 1: The complementary output signal is not determined by the vbusvalid comparator.
[23]	RW	complement	Indicator complement 0: The PHY does not invert the external Vbus indicator signal. 1: The PHY inverts the external Vbus indicator signal.
[22]	RW	termseldlpulse	SRP line pulse drive 0: utmi_txvalid drives the line pulse. 1: utmi_termsel drives the line pulse.
[21:18]	-	reserved	Reserved
[17]	RW	ulpifsls	ULPI FS/LS select 0: ULPI interface 1: ULPI FS/LS serial interface Note: Set GUSBCFG[ulpiumtisel] to 1 before setting this bit.
[16]	-	reserved	Reserved
[15]	RW	phylpwrclkSEL	PHY low-power clock select 0: internal 480 MHz PLL clock 1: external 48 MHz clock
[14]	-	reserved	Reserved



[13:10]	RW	usbtrdtim	USB turnaround time 0101: This field is set to this value when the MAC interface is a 16-bit UTMI+ interface. 1001: This field is set to this value when the MAC interface is an 8-bit UTMI+ interface Other values: reserved
[9]	RW	hnpcap	HNP enable 0: disabled 1: enabled
[8]	RW	srpcap	SRP enable 0: disabled 1: enabled
[7]	-	reserved	Reserved
[6]	R/RW	physel	USB 2.0 high-speed PHY or USB 1.1 full-speed serial transceiver 0: USB 2.0 high-speed UTMI+ or ULPI PHY interface 1: USB 1.1 full-speed serial transceiver If the USB 1.1 full-speed serial transceiver is not selected, this bit is always 0 and write-only. If the USB 2.0 high-speed PHY is not selected, this bit is always 1 and read-only. If the preceding interface types are not selected (non-zero value), this bit is used to select the interface to be activated, and this bit can be read and written.
[5]	R/RW	fsintf	Full-speed serial interface select 0: 6-pin unidirectional full-speed serial interface 1: 3-pin bidirectional full-speed serial interface If the USB 1.1 full-speed serial transceiver is not selected, this bit is always 0 and write-only. If the USB 1.1 full-speed interface is selected, this bit can be used to select the 3-pin or 6-pin interface and can be read and written.
[4]	R/RW	ulpi_utmi_sel	ULPI or UTMI select 0: UTMI+ interface 1: ULPI interface
[3]	R/RW	phyif	PHY interface 0: 8 bits 1: 16 bits



[2:0]	RW	toutcal	HS/FS timeout calibration High-speed operations: One 30-MHz PHY clock = 16 bit times One 60-MHz PHY clock = 8 bit times Full speed operation: One 30-MHz PHY clock = 0.4 bit times One 60-MHz PHY clock = 0.2 bit times One 48-MHz PHY clock = 0.25 bit times
-------	----	---------	--

GRSTCTL

GRSTCTL is a reset hardware feature register.

	Offset Address 0x0010																Register Name GRSTCTL								Total Reset Value 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	ahbidle	dmareq	reserved																txfnum				txffish	rxffish	reserved	frmentrst	reserved	esfirst				
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		
Bits	Access	Name		Description																												
[31]	RO	ahbidle		AHB master idle, indicating whether the AHB master state machine is idle																												
[30]	RO	dmareq		DMA request signal, indicating that the DMA request is in progress. This bit is only for debugging.																												
[29:11]		reserved		Reserved																												



[10:6]	RW	txfnum	<p>TX FIFO ID</p> <p>0x0:</p> <ul style="list-style-type: none">• In host mode, non-periodic TX FIFOs are flushed.• During shared FIFO operations, non-periodic TX FIFOs are flushed in device mode.• In dedicated FIFO mode, TX FIFO 0 is flushed in device mode. <p>0x1:</p> <ul style="list-style-type: none">• In host mode, periodic TX FIFOs are flushed.• During shared FIFO operations, periodic TX FIFO 1 is flushed in device mode.• In dedicated FIFO mode, TX FIFO 1 is flushed in device mode. <p>0x2:</p> <ul style="list-style-type: none">• During shared FIFO operations, periodic TX FIFO 2 is flushed in device mode.• In dedicated FIFO mode, TX FIFO 2 is flushed in device mode. <p>...</p> <p>0xF:</p> <ul style="list-style-type: none">• During shared FIFO operations, periodic TX FIFO 15 is flushed in device mode.• In dedicated FIFO mode, TX FIFO 15 is flushed in device mode. <p>0x10:</p> <ul style="list-style-type: none">• All TX FIFOs are flushed in device or host mode.
[5]	R_WS_S_C	txfflsh	<p>TX FIFO flush</p> <p>This bit can be used to refresh one or all TX FIFOs but FIFOs cannot be refreshed during transaction.</p>
[4]	RO	rxfflsh	<p>RX FIFO flush</p> <p>This bit can be used to refresh all RX FIFOs.</p>
[3]	-	reserved	Reserved
[2]	RO	frmcntrrst	<p>Host frame counter reset</p> <p>This bit is used to reset the frame number counter. When the frame number counter is reset, the number of SOF frames is 0.</p>
[1]	RO	reserved	Reserved



[0]	RW	csftrst	<p>Core soft interrupt</p> <p>1) All interrupts and the CSR register excluding the following bits are cleared:</p> <ul style="list-style-type: none">-PCGCCTL.RstPdwnModule-PCGCCTL.GateHclk-PCGCCTL.PwrClmp-PCGCCTL.StopPPhyLPwrClkSelclk-GUSBCFG.PhyLPwrClkSel-GUSBCFG.DDRSel-GUSBCFG.PHYSel-GUSBCFG.FSIntf-GUSBCFG.ULPI_UTMI_Sel-GUSBCFG.PHYIf-GUSBCFG.TxEndDelay-GUSBCFG.TermSelDLPulse-GUSBCFG.ULPIClkSusM-GUSBCFG.ULPIAutoRes-GUSBCFG.ULPIFsLs-GPIO-GPWRDN-GADPCTL-HCFG.FSLSPclkSel-DCFG.DevSpd-DCTL.SftDiscon <p>2) All module state machines are reset to the idle status, and all TX FIFOs and RX FIFOs are flushed.</p> <p>3) All transactions on the AHB master are interrupted as soon as possible, and all USB transactions are interrupted immediately.</p> <p>4) When the hibernation or ADP feature is enabled, the PMU module is not soft-reset by the core.</p>
-----	----	---------	---

GINTSTS

GINTSTS is a system interrupt register.



Bit	Offset Address 0x0014																Register Name GINTSTS								Total Reset Value 0x0800_0080							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	wkupint	sessreqint	disconnint	conidstschg	lpm_int	ptxfemp	hchint	prtint	residet	fetsusp	incomplpincompisout	incomplpincompisoin	oepint	iepint	epnis	rstrdoneint	eopf	isooutdrop	enumdone	usbrst	usbsusp	erly/susp	reserved	goutnakeff	ginnakeff	ntxtemp	rxflvl	sof	ogint	modenis	curmod	
Reset	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	0	0	0	0	0	
Bits	Access	Name		Description																												
[31]	RWSC	wkupint		Suspend/Resume detection interrupt. Setting this bit to 1 clears the interrupt.																												
[30]	RWSC	sessreqint		In host mode, an interrupt is generated when a device session request is detected. In device mode, an interrupt is generated when utmisrp_bvalid is 1. Setting this bit to 1 clears the interrupt.																												
[29]	RWSC	disconnint		An interrupt is generated when the system detects that the device is disconnected. Setting this bit to 1 clears the interrupt.																												
[28]	RWSC	conidstschg		An interrupt is generated when the slot ID changes. Setting this bit to 1 clears the interrupt.																												
[27]	RWSC	lpm_int		In device mode, an interrupt is generated when the device receives an LPM transaction and transmits a response indicating no errors. In host mode, an interrupt is generated when the device receives an LPM transaction and transmits a response indicating no errors, or when the host core completes the preset LPM transactions. Note: This bit is valid only when LPMC capable or OTG_ENABLE_LPM is 1. Setting this bit to 1 clears the interrupt.																												
[26]	RO	ptxfemp		An interrupt is generated when the periodic TX FIFO is empty or half empty.																												
[25]	RO	hchint		In host mode, an interrupt is generated in a core channel. This bit is cleared by clearing the corresponding status bit.																												
[24]	RO	prtint		In host mode, an interrupt is generated when the status of a port of the DWC_otg core changes. Note: This bit is cleared by clearing the corresponding bit in the host port control and status register.																												



[23]	RWSC	resetdet	In device mode, an interrupt is generated when the device is suspended and the system detects that the USB module is reset in power-down mode. In host mode, this interrupt is not generated. Note: Setting this bit to 1 clears the interrupt.
[22]	RWSC	fetsusp	This interrupt is valid only in DMA mode. This interrupt indicates that the core stops obtaining data from IN endpoints.
[21]	RWSC	incomplpincompisoout	In host mode, an interrupt is generated if there are pending transactions. In device mode, an interrupt is generated when there are isochronous OUT transfers to be completed. Note: Setting this bit to 1 clears the interrupt.
[20]	RWSC	incompisoin	An interrupt is generated when there are isochronous IN transfers to be completed. In scatter/gather DMA mode, this interrupt is not generated. Note: Setting this bit to 1 clears the interrupt.
[19]	RO	oepint	In device mode, an interrupt is generated on an OUT endpoint. This interrupt is cleared by clearing the corresponding status bit in the DOEPINT n register.
[18]	RO	iepint	In device mode, an interrupt is generated on an IN endpoint. This interrupt is cleared by clearing the corresponding status bit in the DOEPINT n register.
[17]	RO	epmis	An interrupt is generated when endpoints mismatch. This bit is valid only during shared FIFO operations. Note: Setting this bit to 1 clears the interrupt.
[16]	RWSC	rstrdoneint	An interrupt is caused by the restore command after hibernation. Note: This bit is valid only when the hibernation function is enabled.
[15]	RWSC	eopf	Periodic frame end interrupt Setting this bit to 1 clears the interrupt.
[14]	RWSC	isooutdrop	Isochronous OUT packet drop interrupt Note: Setting this bit to 1 clears the interrupt.
[13]	RWSC	enumdone	Speed enumeration completion interrupt Note: Setting this bit to 1 clears the interrupt.
[12]	RWSC	usbrst	USB reset detection interrupt Note: Setting this bit to 1 clears the interrupt.
[11]	RWSC	usbsusp	USB suspend detection interrupt Note: Setting this bit to 1 clears the interrupt.
[10]	RWSC	erlysusp	Interrupt indicating that the USB is idle for 3 ms Note: Setting this bit to 1 clears the interrupt.
[9:8]	-	reserved	Reserved



[7]	RO	goutnakeff	An interrupt is generated when the global OUT NAK bit in the device control register is set. This interrupt is cleared by clearing the global OUT NAK bit.
[6]	RO	ginnakeff	An interrupt is generated when the global non-periodic IN NAK is set in the device control register.
[5]	RO	nptxfemp	An interrupt is generated when the non-periodic TX FIFO is empty or half empty.
[4]	RO	rxflvl	An interrupt is generated when there is at least one pending packet to be read by the RX FIFO.
[3]	RWSC	sof	In host mode, an interrupt is generated when an SOF, micro-SOF, or keep-active is being transmitted. In device mode, an interrupt is generated when an SOF token is received by the USB module.
[2]	RO	otgint	An interrupt is generated when an OTG even is in progress. This interrupt is cleared by clearing the corresponding bit in the GOTGINT register.
[1]	RWSC	modemis	Mode mismatch interrupt When the core works in device mode, a register in host mode needs to be accessed. When the core works in host mode, a register in device mode needs to be accessed. Note: Setting this bit to 1 clears the interrupt.
[0]	RO	curmod	Current operating mode 0: device mode 1: host mode

GINTMSK

GINTMSK is a system interrupt mask register.



Bit	Offset Address 0x0018												Register Name GINTMSK												Total Reset Value 0x0000_0000											
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	wkupintmsk	sessreqintmsk	disconnintmsk	conidstschngmsk	lpm_intmsk	ptxfempmsk	hchintmsk	prtintmsk	resetdetmsk	fetsuspmesk	incomplpmkskincompisoutmsk	incompoimnmsk	oepintmsk	iepintmsk	epmismsk	reserved	eopfmsk	isooutdropmsk	enumdonemsk	usbriumsk	usbupmsk	erlysuspmesk	i2cintmsk	ulpickintmsk	goutnakeffmsk	ginnakeffmsk	npxfempmsk	rxflvmsk	sofmsk	oigimmsk	modemismsk	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]	RW	wkupintmsk		Resume/Remote wakeup detection interrupt mask 0: masked 1: not masked																																
[30]	RW	sessreqintmsk		Session request/new session detection interrupt mask 0: masked 1: not masked																																
[29]	RW	disconnintmsk		Disconnect detection interrupt mask. 0: masked 1: not masked																																
[28]	RW	conidstschngmsk		Connector ID status change interrupt mask 0: masked 1: not masked																																
[27]	RW	lpm_intmsk		LPM transaction RX interrupt mask 0: masked 1: not masked																																
[26]	RW	ptxfempmsk		Periodic TX FIFO empty interrupt mask 0: masked 1: not masked																																
[25]	RW	hchintmsk		Host channel interrupt mask 0: masked 1: not masked																																
[24]	RW	prtintmsk		Host port interrupt mask 0: masked 1: not masked																																



[23]	RW	resetdetmsk	Reset detection interrupt mask 0: masked 1: not masked
[22]	RW	fetsuspmask	Data fetch suspended interrupt mask 0: masked 1: not masked
[21]	RW	incomplpmkincom pisououtmsk	Incomplete periodic transfer interrupt mask 0: masked 1: not masked Incomplete isochronous OUT transfer interrupt mask 0: masked 1: not masked
[20]	RW	incompisoimask	Incomplete isochronous IN transfer interrupt mask 0: masked 1: not masked This bit can be enabled only when periodic endpoints are enabled in dedicated TX FIFO mode.
[19]	RW	oepintmsk	OUT endpoints interrupt mask 0: masked 1: not masked
[18]	RW	iepintmsk	IN endpoints interrupt mask 0: masked 1: not masked
[17]	RW	epmismsk	Endpoint mismatch interrupt mask 0: masked 1: not masked
[16]	-	reserved	Reserved
[15]	RW	eopfmsk	End of periodic frame interrupt mask 0: masked 1: not masked
[14]	RW	isooutdropmsk	Device only isochronous OUT packet drop interrupt mask 0: masked 1: not masked
[13]	RW	enumdonemsk	Enumeration done interrupt mask 0: masked 1: not masked



[12]	RW	usbrstmsk	USB reset interrupt mask 0: masked 1: not masked
[11]	RW	usbsuspmsk	USB suspend interrupt mask 0: masked 1: not masked
[10]	RW	erlysuspmsk	Early suspend interrupt mask 0: masked 1: not masked
[9]	RW	i2cintmsk	I ² C interrupt mask 0: masked 1: not masked
[8]	RW	ulpickintmsk	ULPI Carkit interrupt mask 0: masked 1: not masked I ² C Carkit interrupt mask 0: masked 1: not masked
[7]	RW	goutnakeffmsk	Global OUT NAK effective interrupt mask 0: masked 1: not masked
[6]	RW	ginnakeffmsk	Global non-periodic IN NAK effective interrupt mask 0: masked 1: not masked
[5]	RW	nptxfempmsk	Non-periodic TX FIFO empty interrupt mask 0: masked 1: not masked
[4]	RW	rxflvlmsk	Receive FIFO non-empty interrupt mask 0: masked 1: not masked
[3]	RW	sofmsk	Start of (micro)frame interrupt mask 0: masked 1: not masked
[2]	RW	otgintmsk	Device interrupt mask 0: masked 1: not masked



[1]	RW	modemismsk	Mode mismatch interrupt mask 0: masked 1: not masked
[0]	RW	reserved	Reserved

GRXSTSR

GRXSTSR is an RX status debugging read register.

	Offset Address 0x001C																Register Name GRXSTSR								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved_devicemode				fn_devicemode				pktsts_devicemode				dpid_devicemode				bcnt_devicemode				chnum_device 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			
Bits	Access	Name			Description																											
[31:25]	-	reserved_devicemode			Reserved																											
[24:21]	RO	fn_devicemode			Number of frames																											
[20:17]	RO	pktsts_devicemode			Data packet status 0001: global OUT NAK (interrupt triggered) 0010: IN data packet received 0111: channel stop (interrupt triggered) Other values: reserved																											
[16:15]	RO	dpid_devicemode			Data PID of the received packet 00: DATA0 10: DATA1 01: DATA2 11: MDATA																											
[14:4]	RO	bcnt_devicemode			Byte size of the received IN data packets																											



[3:0]	RO	chnum_devicemode	Number of channels from which the current data packets are received
-------	----	------------------	---

GRXSTSP

GRXSTSP is an RX status read&pop register.

	Offset Address										Register Name										Total Reset Value											
	0x0020										GRXSTSP										0x0000_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved_devicemode					fn_devicemode					pktsts_devicemode					dpid_devicemode					bcnt_devicemode					chnum_devicemode						
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved_devicemode			Reserved																											
[24:21]	RO	fn_devicemode			Number of frames																											
[20:17]	RO	pktsts_devicemode			Data packet status 0001: global OUT NAK (interrupt triggered) 0010: IN data packet received 0111: IN transfer completion (interrupt triggered) 0101: data toggle error (interrupt triggered) 0111: channel stop (interrupt triggered) Other values: reserved																											
[16:15]	RO	dpid_devicemode			Data PID of the received packet 00: DATA0 10: DATA1 01: DATA2 11: MDATA																											
[14:4]	RO	bcnt_devicemode			Byte size of the received IN data packets																											
[3:0]	RO	chnum_devicemode			Number of channels from which the current data packets are received																											



GRXFSIZ

GRXFSIZ is an RX FIFO size configuration register.

Offset Address								Register Name								Total Reset Value																
0x0024								GRXFSIZ								0x0000_0211																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																rxfdep															
Reset	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	1	
Bits	Access	Name			Description																											
[31:16]	RO	reserved			Reserved																											
[15:0]	R/RW	rxfdep			RX FIFO depth 16–32, 768																											

GNPTXFSIZ

GNPTXFSIZ is a non-periodic TX FIFO size configuration register.

Offset Address								Register Name								Total Reset Value																
0x0028								GNPTXFSIZ								0x0100_0211																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ineptxf0dep																ineptxf0staddr															
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1		
Bits	Access	Name			Description																											
[31:16]	RW	ineptxf0dep			Depth of IN endpoint TX FIFO 0 16–32, 768																											
[15:0]	RW	ineptxf0staddr			Start address for the TX RAM of IN endpoint FIFO 0																											

GNPTXSTS

GNPTXSTS is a non-periodic TX FIFO and non-periodic TX request query register.



Offset Address																Register Name								Total Reset Value										
0x002C																GNPTXSTS								0x0008_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	nptxqtop																nptxspcavail																nptxfspcavail
Reset	0	0	0	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	0	0			
Bits	Access	Name			Description																													
[31]	-	reserved			Reserved																													
[30:24]	RO	nptxqtop			Non-periodic TX request queue top Bit[30:27]: number of channels or endpoints bit[26:25]: 00: IN/OUT token 01: zero-length TX packet (device IN/host OUT) 10: PING/CSPLIT token 11: channel stop command Bit[24]: stop (last attempt on the selected channel or endpoint)																													
[23:16]	RO	nptxspcavail			Available space for the non-periodic TX request queue 0x0: The non-periodic TX request queue is full. 0x1: One address is available. 0x2: Two addresses are available. 0xn: n addresses ($0 \leq n \leq 8$) are available. Other values: reserved																													
[15:0]	RO	nptxfspcavail			Total available space for the non-periodic TX request queue 0x0: The non-periodic TX FIFO is Full. 0x1: One word is available. 0x2: Two words are available. 0xn: n words ($0 \leq n \leq 32768$) are available. 0x8000: 32768 words are available. Other values: reserved																													

GI2CCTL

GI2CCTL is an I²C access register.



	Offset Address																Register Name								Total Reset Value							
	0x0030																GI2CCTL								0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	bsydone	rw	reserved	i2cdatse0	i2cdevadr	i2csuspcctl	ack	i2cen	addr				regaddr				rwdata															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bits	Access	Name				Description																										
[31]	RWSC	bsydone				I^2C busy/done 0: I^2C done 1: I^2C busy																										
[30]	RW	rw				Read/Write indicator 0: write 1: read																										
[29]	RW	reserved				Reserved																										
[28]	RW	i2cdatse0				I^2C mode of the FS interface 0: VP_VM USB mode 1: DAT_SE0 USB mode																										
[27:26]	RW	i2cdevadr				I^2C device address 00: 7'h2C 01: 7'h2D 10: 7'h2E 11: 7'h2F																										
[25]	RW	i2csuspcctl				I^2C suspend control 0: utmi_suspend_n is used. 1: The suspend bit in the PHY register is edited by using the I^2C write function.																										
[24]	RO	ack				I^2C ACK 0: NAK 1: ACK																										
[23]	RW	i2cen				I^2C transaction enable 0: disabled 1: enabled																										
[22:16]	RW	addr				I^2C slave address																										
[15:8]	RW	regaddr				I^2C register address																										
[7:0]	RW	rwdata				I^2C read/write data																										



GPVNDCTL

GPVNDCTL is a PHY vendor control register.

Bit	Offset Address 0x0034																Register Name GPVNDCTL								Total Reset Value 0x0000_0000							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	vstsdone	vstsbsy	newreq	reserved	regwr	regaddr	vctrlextregaddr								regdata																
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RWSC	vstsdone		Vendor control access (done) 0: The new register request is set. 1: Vendor control access ends.																												
[26]	RO	vstsbsy		Vendor control access (busy) 0: Vendor control access ends. 1: Vendor control access is in progress.																												
[25]	RWSC	newreq		New vendor control access 0: no access request 1: new vendor control access request																												
[24:23]	-	reserved		Reserved																												
[22]	RW	regwr		Register read/write 0: read 1: write																												
[21:16]	RW	regaddr		Register access address																												
[15:8]	RW	vctrlextregaddr		Address for the UTMI+vendor control register Bit[15:12]: 4-bit parallel output bus addressing Bit[11:8]: from utmi_vcontrol[3:0]																												
[7:0]	RW	regdata		Register data This bit is valid after the Vstatus done bit is set.																												



GGPIO

GGPIO is a GPIO register.

GUID

GUID is a user ID register.

GSNPSID

GSNPSID is a Synopsys ID query register.

Offset Address								Register Name								Total Reset Value																
0x0040								GSNPSID								0x4F54_300A																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	synopsysid																															
Reset	0	1	0	0	1	1	1	1	0	1	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1	0	
Bits	Access	Name			Description																											
[31:0]	RW	synopsysid			ID of the current DWC_otg core																											



GHWCFG1

GHWCFG1 is user hardware configuration register 1.

GHWCFG2

GHWCFG2 is user hardware configuration register 2.



Offset Address 0x0048												Register Name GHWCFG2												Total Reset Value 0x2284_C850											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	otg_enable_ic_usb	tknqdepth				ptxqdepth	nptxqdepth	reserved	multiprocintrpt	dynfifosizing	periosupport	numhstchnl				numdeveps				fphystype	hsphystype	singpnt	otgarch	otgmode											
Reset	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0			
Bits	Access	Name		Description																															
[31]	RO	otg_enable_ic_usb		IC_USB select enable 0: disabled 1: enabled																															
[30:26]	RO	tknqdepth		Depth of the IN token sequence learning queue in device mode The value range is 0–30.																															
[25:24]	RO	ptxqdepth		Depth of the periodic request queue in host mode 00: 2 01: 4 10: 8 11: 16																															
[23:22]	RO	nptxqdepth		Depth of the non-periodic request queue 00: 2 01: 4 10: 8 Other values: reserved																															
[21]		reserved		Reserved																															
[20]	RW	multiprocintrpt		Multi-processor interrupt enable 0: disabled 1: enabled																															
[19]	RO	dynfifosizing		FIFO size dynamic change enable 0: disabled 1: enabled																															
[18]	RO	periosupport		Periodic OUT channel support in host mode 0: not supported 1: supported																															



[17:14]	RO	numhstchnl	Number of host channels 0–15: The value 0 indicates one channel, and the value 15 indicates 16 channels.
[13:10]	RO	numdeveps	Number of device endpoints (excluding endpoint 0) The value range is 0–15.
[9:8]	RO	fsphytype	Type of the full-speed PHY interface 00: The full-speed interface is not supported. 01: The full-speed interface is supported. 10: FS pins share with UTMI+ pins. 11: FS pins share with ULPI pins.
[7:6]	RO	hsphytype	Type of the high-speed PHY interface 00: high-speed interface not supported 01: UTMI+ 10: ULPI 11: UTMI+ and ULPI
[5]	RO	singpnt	Point-to-point 0: multi-point application (hub and split supported) 1: single-point application (hub and split not supported)
[4:3]	RO	otgarch	Device architecture 00: slave-only 01: external DMA 10: internal DMA Other values: reserved
[2:0]	RO	otgmode	Operating mode 000: HNP- and SRP-capable OTG (host & device) 001: SRP-Capable OTG (host & device) 010: Non-HNP and Non-SRP Capable OTG (host & device) 011: SRP-capable device 100: Non-OTG device 101: SRP-capable host 110: Non-OTG host Other values: reserved

GHWCFG3

GHWCFG3 is user hardware configuration register 3.



	Offset Address 0x004C														Register Name GHWCFG3										Total Reset Value 0x0501_54E8									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	dffifodepth																																	
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	1	1	1	0	1	0	0	0		
Bits	Access	Name	Description																															
[31:16]	RO	dffifodepth	DFIFO depth 32–32, 768																															
[15]	RO	lpemode	LPM mode select																															
[14]	RO	bcsupport	Whether the HS device controller supports the battery charger 0: not supported 1: supported																															
[13]	RO	hsicmode	HSIC mode select 0: The HSIC-capable shares with the UTMI PHY interface 1: The HSIC mode is not selected.																															
[12]	RO	adpsupport	Whether the device controller has an ADP logic 0: no 1: yes																															
[11]	RO	rsttype	Reset mode of clock always blocks 0: asynchronous reset 1: synchronous reset																															
[10]	RO	optfeature	Feature removal Features include the user ID register, GPIO interface ports, and SOF toggle and counter ports. 0: not removed 1: removed																															
[9]	RO	vndctlupt	Vendor control interface support 0: not supported 1: supported																															
[8]	RO	i2cintsel	I ² C interface select 0: not selected 1: selected																															



[7]	RO	otgen	Device function enable 0: disabled 1: enabled
[6:4]	RO	pktsizewidth	Data packet width 000: 4 bits 001: 5 bits 010: 6 bits 011: 7 bits 100: 8 bits 101: 9 bits 110: 10 bits Other values: reserved
[3:0]	RO	xfersizewidth	Transfer width 0000: 11 bits 0001: 12 bits 1000: 19 bits Other values: reserved

GHWCFG4

GHWCFG4 is user hardware configuration register 4.

Offset Address												Register Name												Total Reset Value											
0x0050												GHWCFG4												0x4600_8020											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	descdma	descdmaen	ineps				dedfifomode	sessendfltr	bvalidfltr	avalidfltr	vbusvalidfltr	iddgfltr	numctleps		phydatawidth	reserved						extndedhibernation	hibernation	ahbfreq	partialpwrdn	numdevperioeps									
Reset	0	1	0	0	0	1	1	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]	RO		descdma				Scatter/Gather DMA 0: no dynamic configuration 1: dynamic configuration																												
[30]	RO		descdmaen				Pad slew rate 0: fast 1: slow																												



[29:26]	RO	ineps	Number of IN endpoints in device mode 0: 1 1: 2 ... 15: 16
[25]	RO	dedfifomode	Dedicated TX FIFO enable for device IN endpoints 0: disabled 1: enabled
[24]	RO	sessendfltr	session_end filter enable 0: disabled 1: enabled
[23]	RO	bvalidfltr	b_valid filter enable 0: disabled 1: enabled
[22]	RO	avalidfltr	a_valid filter enable 0: disabled 1: enabled
[21]	RO	vbusvalidfltr	Vbus valid filter enable 0: disabled 1: enabled
[20]	RO	iddgfltr	IDDIG filter enable 0: disabled 1: enabled
[19:16]	RO	numctleps	Number of controlled endpoints excluding endpoint 0 in device mode The value range is 0–15.
[15:14]	RO	phydatawidth	Data width of the UTMI+ PHY/ULPI-to-internal UTMI+ wrapper 00: 8 bits 01: 16 bits 10: 8/16 bits (configurable by software) Other values: reserved
[13:8]	RO	reserved	Reserved
[7]	RO	extndedhibernation	Extended hibernation enable 0: disabled 1: enabled
[6]	RO	hibernation	Hibernation enable 0: disabled 1: enabled



[5]	RO	ahbfreq	Whether the minimum AHB frequency is lower than 60 MHz 0: no 1: yes
[4]	RO	partialpwrdsn	Partial power-down enable 0: disabled 1: enabled
[3:0]	RO	numdevperioeps	Number of periodic IN endpoints in device mode The value range is 0–15.

GLPMCFG

GLPMCFG is an LPM configuration register.

Bit	Offset Address 0x0054																Register Name GLPMCFG								Total Reset Value 0x0000_0000							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	invselhsic	reserved	rstrslpst	enbesl	lpm_retrycnt_sts	sndlpn	lpm_retry_cnt	lpm_chnl_indx	lresumeok	slpst	core1res	hird_thres	enblslpm	bremotewake	hird	applres	lpmcap															
Reset	0	0	0	0	0	0	0	0	0	0	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	invselhsic				HSIC enable (by using HSIC-invert) If if_sel_hsic is 1: InvSelHsic = 1: disabled InvSelHsic = 0: enabled If if_sel_hsic is 0: InvSelHsic = 1: enabled InvSelHsic = 0: disabled																										
[30]	-	reserved				Reserved																										
[29]	RW	rstrslpst				Restore SlpSts 0: The core enters the shallow sleep mode. 1: The core enters the deep sleep mode.																										
[28]	RW	enbesl				Best effort service latency (BESL) enable 0: disabled 1: enabled																										



[27:25]	RO	lpm_retrycnt_sts	LPM retry count This field is used to count the remaining retries.
[24]	RWSC	sndlpm	LPM transaction TX
[23:21]	RW	lpm_retry_cnt	LPM retry count statistics When a device provides error information, this field shows the extra LPM retries of the host before the device validity signal arrives.
[20:17]	RW	lpm_chnl_indx	LPM channel index The number of LPM transaction channels is provided. The core automatically calculates the device address and number of endpoints based on the LPM channel index.
[16]	RO	l1resumeok	Resume from the sleep state 0: resumed 1: not resumed
[15]	RO	slpst	Port sleep state 0: resumed 1: sleep
[14:13]	RO	corel1res	LPM response In device mode, this field reflects the response to the LPM transaction. In host mode, this field indicates the handshake response from the device during LPM transaction. 11: ACK 10: NYET 01: STALL 00: ERROR (no handshake response)
[12:8]	RW	hird_thres	BESL or HIRD threshold
[7]	RW	enblslpm	utmi_sleep_n enable 0: disabled 1: enabled
[6]	RW	bremotewake	Remote wakeup enable In host mode, the enable signal is transmitted from the wIndex field of the LPM transaction. In device mode (read-only), when the ACK, NYET, or STALL response is transmitted to the LPM transaction, this bit is updated together with the received LPM token, bRemoteWake, and bmAttribute.



[5:2]	RW	hird	<p>When EnBESL is 0: Host-initiated resume period. In host mode, this field is assigned by the LPM transaction. In device mode, this field is updated by LPM token HIRD bmAttribute.</p> <p>S1. No HIRD[3:0] THIRD (μs)</p> <table><tbody><tr><td>1</td><td>4'b0000</td><td>50</td></tr><tr><td>2</td><td>4'b0001</td><td>125</td></tr><tr><td>3</td><td>4'b0010</td><td>200</td></tr><tr><td>4</td><td>4'b0011</td><td>275</td></tr><tr><td>5</td><td>4'b0100</td><td>350</td></tr><tr><td>6</td><td>4'b0101</td><td>425</td></tr><tr><td>7</td><td>4'b0110</td><td>500</td></tr><tr><td>8</td><td>4'b0111</td><td>575</td></tr><tr><td>9</td><td>4'b1000</td><td>650</td></tr><tr><td>10</td><td>4'b1001</td><td>725</td></tr><tr><td>11</td><td>4'b1010</td><td>800</td></tr><tr><td>12</td><td>4'b1011</td><td>875</td></tr><tr><td>13</td><td>4'b1100</td><td>950</td></tr><tr><td>14</td><td>4'b1101</td><td>1025</td></tr><tr><td>15</td><td>4'b1110</td><td>1100</td></tr><tr><td>16</td><td>4'b1111</td><td>1175</td></tr></tbody></table> <p>When EnBESL is 1: BESL In host mode, the BESL value is transmitted to the LPM transaction In device mode, this field is updated by LPM token BESL bmAttribute.</p>	1	4'b0000	50	2	4'b0001	125	3	4'b0010	200	4	4'b0011	275	5	4'b0100	350	6	4'b0101	425	7	4'b0110	500	8	4'b0111	575	9	4'b1000	650	10	4'b1001	725	11	4'b1010	800	12	4'b1011	875	13	4'b1100	950	14	4'b1101	1025	15	4'b1110	1100	16	4'b1111	1175
1	4'b0000	50																																																	
2	4'b0001	125																																																	
3	4'b0010	200																																																	
4	4'b0011	275																																																	
5	4'b0100	350																																																	
6	4'b0101	425																																																	
7	4'b0110	500																																																	
8	4'b0111	575																																																	
9	4'b1000	650																																																	
10	4'b1001	725																																																	
11	4'b1010	800																																																	
12	4'b1011	875																																																	
13	4'b1100	950																																																	
14	4'b1101	1025																																																	
15	4'b1110	1100																																																	
16	4'b1111	1175																																																	
[1]	RW	appl1res	LPM response 0: NYET 1: ACK																																																
[0]	RW	lpmcap	LPM enable 0: disabled 1: enabled																																																

GPWRDN

GPWRDN is a power-down register.



Offset Address 0x0058												Register Name GPWRDN												Total Reset Value 0x1320_0010											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	reserved	multvalidbc					adpint	bsessvld	iddig	linestate			stschngintmsk	stschngint	srpdetectmsk	srpdetect	reserved								disablebus	pwrndnswitch	pwrdrst_n	pwrndnclmp	reserved	pmuactv	pmuintsel				
Reset	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0					
Bits	Access	Name		Description																															
[31:29]	-	reserved		Reserved																															
[28:24]	RO	multvalidbc		BC ACA input Bit 28: rid_float Bit 27: rid_gnd Bit 26: rid_a Bit 25: rid_b Bit 24: rid_c																															
[23]	RWSC	adpint		An interrupt is generated when the ADP is in progress Setting this bit to 1 clears the interrupt.																															
[22]	RO	bsessvld		B session validity 0: B-Valid = 0 1: B-valid = 1																															
[21]	RO	iddig		IDDIG signal status indicator Current operating mode 0: host mode 1: device mode																															
[20:19]	RO	linestate		Current line state indicator 00: DM = 0, DP = 0 01: DM = 0, DP = 1 10: DM = 1, DP = 0 11: undefined																															
[18]	RW	stschngintmsk		StsChng interrupt mask 0: masked 1: not masked																															



[17]	RWSC	stschgint	StsChng interrupt The bit indicates whether the status of the IDDIG or BSessVld signal changes. 0: not changed 1: changed
[16]	RW	srpdetectmsk	SRP detection interrupt mask
[15]	RWSC	srpdetect	SRP detection 0: not detected 1: detected
[14:7]	-	reserved	Reserved
[6]	RW	disablevbus	Vbus disable In host mode: 0: PrtPwr is not 0. 1: PrtPwr is 0. In device mode: 0: The level of the bvalid signal is high. 1: The level of the bvalid signal is low. This bit is valid only when GPWRDN.PMUActv is 1.
[5]	RW	pwrdnswtch	Power-down switch 0: on 1: off
[4]	RW	pwrdnrst_n	Power-down reset 0: reset 1: not reset
[3]	RW	pwrnclmp	Power-down beat enable 0: disabled 1: enabled
[2]	-	reserved	Reserved
[1]	RW	pmuactv	PMU enable 0: disabled 1: enabled
[0]	RW	pmuintsel	PMU interrupt select 0: internal DWC_otg_core interrupt 1: external DWC_otg_core interrupt

GDFIFO CFG

GDFIFO CFG is a DFIFO software configuration register.



Offset Address																Register Name								Total Reset Value													
Bit	0x005C															GDFIFO CFG							0x0501_0511														
Name	epinfobaseaddr															gdfifocfg																					
Reset	0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 1																																				
Bits	Access	Name															Description																				
[31:16]	RW	epinfobaseaddr															Start address for the EP info controller																				
[15:0]	RW	gdfifocfg															DFIFO size, which is dynamically configured																				

GADPCTL

GADPCTL is an ADP timer control and status register.

Offset Address																Register Name								Total Reset Value							
Bit	0x0060															GADPCTL							0x0000_0000								
Name	reserved	ar	adptoutmsk	adpsnsintmsk	adpprintmsk	adpoutint	adpsnsint	adpprintint	adpen	adptes	enasus	enaprb	rtim															prper	pribdelta	prbdschg	
Reset	0 0																													0 0	
Bits	Access	Name															Description														
[31:29]	RW	reserved															Reserved														
[28:27]	RW	ar															Access request 00: read/write 01: read-only 10: write-only 11: reserved														
[26]	RW	adptoutmsk															ADP timeout interrupt mask 0: masked 1: not masked Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.														
[25]	RW	adpsnsintmsk															ADP sense interrupt mask 0: masked 1: not masked Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.														



[24]	RWC	adpprbintmsk	ADB probe interrupt mask 0: masked 1: not masked Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.
[23]	RWC	adptoutint	ADP timeout interrupt Setting this bit to 1 clears the interrupt. Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.
[22]	RWC	adpsnsint	ADP sense interrupt Setting this bit to 1 clears the interrupt. Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.
[21]	RWC	adpprbint	ADP probe interrupt Setting this bit to 1 clears the interrupt. Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.
[20]	RW	adpen	ADP enable 0: disabled 1: enabled Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.
[19]	RWC	adpres	ADP reset 0: not reset +1: reset Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.
[18]	RW	enasns	Sense enable 0: disabled 1: enabled Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.
[17]	RW	enaprb	Probe enable 0: disabled 1: enabled Note: This bit is valid only when GOTGCTL[20] (OTG_Ver) is 1.
[16:6]	RO	rtime	Ramp time The following is the mapping between the values and 32 kHz clock cycle: 0x000: 1 cycles 0x001: 2 cycles 0x002: 3 cycles ... 0x7FF: 2048 cycles



[5:4]	RW	prbper	Probe period 00: 0.625s to 0.925s (0.775s typically) 01: 1.25s to 1.85s (1.55s typically) 10: 1.9s to 2.6s (2.275s typically) 11: reserved
[3:2]	RW	prbdelta	Probe delta The following is the mapping between the values and 32 kHz clock cycle: 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
[1:0]	RW	prbdschg	Probe discharge TADP_DSCHG time 00: 4 ms 01: 8 ms 10: 16 ms 11: 32 ms

HPTXFSIZ

HPTXFSIZ is a TX FIFO configuration register.

Offset Address								Register Name								Total Reset Value																
0x0100								HPTXFSIZ								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ptxfsiz												ptxfstaddr																			
Reset	0	0	0	0	0	0	0	0	0	0	0	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/RW	ptxfsiz			Depth of the host periodic TX FIFO, ranging from 16 to 32768																											
[15:0]	RO/RW	ptxfstaddr			Start address for the host periodic TX FIFO																											

DPTXFSIZN

DPTXFSIZN is a device periodic TX FIFO-*n* size register.



This register is valid only in shared FIFO mode.



DIEPTXFN

DIEPTXFN is a device IN endpoint TX FIFO size register.



NOTE

This register is valid only in dedicated FIFO mode.

DCFG

DCFG is a device configuration register.



Offset Address								Register Name								Total Reset Value																
0x0800								DCFG								0x8100_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	resvalid				perschintvl	descdma	epmiscnt				reserved				endevotnak	perfrint	devaddr				ena32khzsusp	nzstsouthshk	dvspd									
Reset	1	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		
Bits	Access	Name			Description																											
[31:26]	RW	resvalid			Resume time control This bit is valid only when DCFG[ena32khzsusp] is 1.																											
[25:24]	RW	perschintvl			Ratio of the (micro) frames in scatter/gather DMA mode 00: 25% of (micro)frame 01: 50% of (micro)frame 10: 75% of (micro)frame 11: reserved																											
[23]	RW	descdma			Scatter/Gather DMA enable in device mode 0: disabled 1: enabled																											
[22:18]	RW	epmiscnt			IN endpoint mismatch statistics																											
[17:14]	RW	reserved			Reserved																											
[13]	RW	endevotnak			Device OUT NAK enable 0: disabled 1: enabled																											
[12:11]	RW	perfrint			Periodic frame interval 00: 80% of the (micro)frame interval 01: 85% 10: 90% 11: 95%																											
[10:4]	RW	devaddr			Device address																											
[3]	RW	ena32khzsusp			32 kHz suspend mode enable 0: disabled 1: enabled																											
[2]	RW	nzstsouthshk			Non-zero-length status OUT handshake select																											



[1:0]	RW	devspd	Device speed 00: high speed (30 MHz or 60 MHz USB 2.0 PHY clock) 01: full speed (30 MHz or 60 MHz USB 2.0 PHY clock) 10: low speed (6 MHz USB 1.1 transceiver clock) 11: full speed (48 MHz USB 1.1 transceiver clock)
-------	----	--------	--

DCTL

DCTL is a device control register.

	Offset Address 0x0804																Register Name DCTL								Total Reset Value 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	reserved																encontonbna	nakonbble	ignrfrmnum	gmc	reserved	pwrnrgdone	cgnpmak	sgnmpmaks	sgnppmaks	tstctl	goutnaksts	gnppmaks	sftdiscon	rmtwkpisig		
Reset	0	0	0	0	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																													
[31:18]	RW	reserved	Reserved																													
[17]	RW	encontonbna	Continue on BNA enable 0: disabled 1: enabled																													
[16]	RW	nakonbble	NAK on babble error 0: no operation 1: NAK configured on the received babble																													
[15]	RW	ignrfrmnum	Whether the number of frames on isochronous endpoints is ignored 0: not ignored 1: ignored																													
[14:13]	RW	gmc	Global multi count 0: invalid 01: 1 packet 10: 2 packets 11: 3 packets																													
[12]	RW	reserved	Reserved																													



[11]	RW	pwrонprgdone	Wakeup from the power-down mode 0: not woken up 1: woken up
[10]	WO	cгoutnak	Global OUT NAK clear 0: no operation 1: cleared
[9]	WO	sгoutnak	Global OUT NAK 0: no operation 1: configured
[8]	WO	cgnpinnak	Global non-periodic IN NAK clear 0: no operation 1: cleared
[7]	WO	sgnpinnak	Global non-periodic IN NAK 0: no operation 1: configured
[6:4]	RW	tstctl	Test control 000: test mode disabled 001: Test_J mode 010: Test_K mode 011: Test_SE0_NAK mode 100: Test_Packet mode 101: Test_Force_Enable Other values: reserved
[3]	RO	goutnaksts	Global OUT NAK status 0: handshake TX 1: NAK handshake TX
[2]	RO	gnpinnaksts	Global non-periodic IN NAK status 0: handshake TX 1: NAK handshake TX
[1]	RW	sftdiscon	Soft disconnection 0: common mode 1: The core drives phy_opmode_o to 2'b01, and then triggers USB disconnection.
[0]	RW	rmtwkupsig	Remote wakeup signal TX 0: not woken up remotely 1: woken up remotely



DSTS

DSTS is a device status query register.

	Offset Address										Register Name										Total Reset Value												
	0x0808										DSTS										0x0007_FF02												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved										devlnsts	soffn										reserved	errticerr	enumspd	suspst								
Reset	0	0	0	0	0	0	0	0	0	0		0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0		
Bits	Access	Name			Description																												
[31:24]	RW	reserved			Reserved																												
[23:22]	RO	devlnsts			Logic level of the current USB data line Bit[23]: Logic level of D+ Bit[22]: Logic level of D-																												
[21:8]	RO	soffn			Number of frames or micro frames in the received SOF packets																												
[7:4]		reserved			Reserved																												
[3]	RO	errticerr			Irregular error																												
[2:1]	RO	enumspd			Enumeration speed 00: high speed (30 MHz or 60 MHz PHY clock) 01: full speed (30 MHz or 60 MHz PHY clock) 10: low speed (6 MHz PHY clock). 11: full speed (48 MHz PHY clock).																												
[0]	RO	suspst			Suspend status. When the suspend conditions are detected, this bit is 1.																												

DIEPMSK

DIEPMSK is a common interrupt mask register for device IN endpoints.





[1]	RW	epdisbldmsk	Endpoint disabled interrupt mask 0: masked 1: not masked
[0]	RW	xfercomplmsk	Transfer completion interrupt mask 0: masked 1: not masked

DOEPMSK

DOEPMSK is a common interrupt mask register for device OUT endpoints.

	Offset Address 0x0814																Register Name DOEPMSK								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																nyetmsk	nakmsk	bbleerrmsk	reserved	bnaoutintrmsk	outpktermask	reserved	back2backsetupmsk	sisphservdmsk	outtknepdismsk	setupmsk	ahberrmsk	epdisbldmsk	xfercomplmsk		
Reset	0	0	0	0	0	0	0	0	0	0	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]	RW	reserved	Reserved																													
[14]	RW	nyetmsk	NYET interrupt mask 0: masked 1: not masked																													
[13]	RW	nakmsk	NAK interrupt mask 0: masked 1: not masked																													
[12]	RW	bbleerrmsk	Babble error interrupt mask 0: masked 1: not masked																													
[11:10]		reserved	Reserved																													
[9]	RW	bnaoutintrmsk	BNA interrupt mask 0: masked 1: not masked																													



[8]	RW	outpkterrmsk	OUT packet error interrupt mask 0: masked 1: not masked
[7]	RW	reserved	Reserved
[6]	RW	back2backsetupmsk	Back-to-back setup packets RX interrupt mask 0: masked 1: not masked
[5]	RW	stsphsercvdmsk	Status phase RX interrupt mask 0: masked 1: not masked
[4]	RW	outtknepdismsk	W OUT token RX when endpoint disabled interrupt mask 0: masked 1: not masked
[3]	RW	setupmsk	Setup phase done interrupt mask 0: masked 1: not masked
[2]	RW	ahberrmsk	AHB error interrupt mask 0: masked 1: not masked
[1]	RW	epdisbldmsk	Endpoint disabled interrupt mask 0: masked 1: not masked
[0]	RW	xfercomplmsk	Transfer completion interrupt mask 0: masked 1: not masked

DAINT

DAINT is an interrupt register for all device endpoints.



Offset Address																Register Name								Total Reset Value								
0x0818																DAINT								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	outepint																inepint															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	outepint				OUT endpoint interrupt Bit 16: OUT endpoint 0 ... Bit 31: OUT endpoint 15																										
[15:0]	RO	inepint				IN endpoint interrupt Bit 0: IN endpoint 0 ... Bit 15: IN endpoint 15																										

DAINTMSK

DAINTMSK is an interrupt mask register for all endpoints.

Offset Address																Register Name								Total Reset Value								
0x081C																DAINTMSK								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	outepmsk																inepmsk															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	outepmsk				OUT endpoint interrupt mask Bit 16: OUT endpoint 0 ... Bit 31: OUT endpoint 15																										
[15:0]	RW	inepmsk				IN endpoint interrupt mask Bit 0: IN endpoint 0 ... Bit 15: IN endpoint 15																										

DTKNQR1

DTKNQR1 is device IN token sequence learning queue read register 1.



DTKNQR2

DTKNQR2 is device IN token sequence learning queue read register 2.



DTKNQR3

DTKNQR3 is device IN token sequence learning queue read register 3.

Offset Address								Register Name								Total Reset Value																
0x0830								DTKNQR3								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	eptkn																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	eptkn		Endpoint token Bits[31:28]: Endpoint number of token 21 Bits[27:24]: Endpoint number of token 20 ... Bits[7:4]: Endpoint number of token 15 Bits[3:0]: Endpoint number of token 14																												

DTKNQR4

DTKNQR4 is device IN token sequence learning queue read register 4.

DVBUSDIS

DVBUSDIS is a device Vbus discharge time register.



Offset Address																Register Name								Total Reset Value								
0x0828																DVBUSDIS								0x0000_17D7								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																dvbusdis															
Reset	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	0	1	1	1	
Bits	Access	Name				Description																										
[31:16]	-	reserved				Reserved																										
[15:0]	RW	dvbusdis				Device Vbus discharge time																										

DVBUSPULSE

DVBUSPULSE is a device Vbus pulse time register.

Offset Address																Register Name								Total Reset Value								
0x082C																DVBUSPULSE								0x0000_05B8								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																dvbuspulse															
Reset	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	0	0	0	0		
Bits	Access	Name				Description																										
[31:12]		reserved				Reserved																										
[11:0]	RO	dvbuspulse				Device Vbus pulse time																										

DTHRCTL

DTHRCTL is a device threshold control register.



Offset Address																Register Name								Total Reset Value								
	0x0830															DTHRCTL								0x0C10_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		arbprken	reserved	rxthrlen								rxthren		reserved		ahbthrratio	txthrlen								isothren	nonisothren					
Reset	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		
Bits	Access	Name		Description																												
[31:28]	RW	reserved		Reserved																												
[27]	RW	arbprken		Arbitrator parking enable 0: disabled 1: enabled																												
[26]	RW	reserved		Reserved																												
[25:17]	RW	rxthrlen		RX threshold length																												
[16]	RW	rxthren		RX threshold enable																												
[15:13]	RW	reserved		Reserved																												
[12:11]	RW	ahbthrratio		AHB threshold ratio 00: AHB threshold = MAC threshold 01: AHB threshold = MAC threshold/2 10: AHB threshold = MAC threshold/4 11: AHB threshold = MAC threshold/8																												
[10:2]	RW	txthrlen		TX threshold length																												
[1]	RW	isothren		ISO IN endpoint threshold enable 0: disabled 1: enabled																												
[0]	RW	nonisothren		Non-ISO IN endpoint threshold enable 0: disabled 1: enabled																												

DIEPEMPMSK

DIEPEMPMSK is a FIFO empty interrupt mask register for device IN endpoints.



Offset Address								Register Name								Total Reset Value																
0x0834								DIEPEMPMSK								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																ineptxfempmsk															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	ineptxfempmsk			IN EP TX FIFO empty interrupt mask Bit 0: IN endpoint 0 ... Bit 15: IN endpoint 15																											

DEACHINT

DEACHINT is an interrupt register for each device endpoint.

Offset Address								Register Name								Total Reset Value																
0x0838								DEACHINT								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	echoutepint												echinepint																			
Reset	0	0	0	0	0	0	0	0	0	0	0	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	echoutepint			OUT endpoint interrupt Bit 16 controls OUT endpoint 0. ... Bit 31 controls OUT endpoint 15.																											
[15:0]	RW	echinepint			IN endpoint interrupt Bit 15 controls IN endpoint 0. ... Bit 0 controls IN endpoint 15.																											

DEACHINTMSK

DEACHINTMSK is an interrupt mask register for each device endpoint.



Offset Address								Register Name								Total Reset Value																
0x083C								DEACHINTMSK								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	echoutepmsk																echinepmsk															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	echoutepmsk				OUT endpoint interrupt mask Bit 16 controls OUT endpoint 0. ... Bit 31 controls OUT endpoint 15. 0: masked 1: not masked																										
[15:0]	RW	echinepmsk				IN Endpoint interrupt mask Bit 16 controls IN endpoint 0. ... Bit 31 controls IN endpoint 15. 0: masked 1: not masked																										

DIEPEACHMSKN

DIEPEACHMSKN is an interrupt register for device IN endpoint n .



[13]	RW	nakmsk	NAK interrupt mask 0: masked 1: not masked
[12:10]	-	reserved	Reserved
[9]	RW	bmainintrmsk	BNA interrupt mask 0: masked 1: not masked
[8]	RW	txfifoundrnmsk	FIFO underrun interrupt mask 0: masked 1: not masked
[7]	-	reserved	Reserved
[6]	RW	inepnakeffmsk	IN endpoint NAK effective interrupt mask 0: masked 1: not masked
[5]	RW	intknepmismsk	IN token received with EP mismatch interrupt mask 0: masked 1: not masked
[4]	RW	intkntxfempmsk	IN token received when TX FIFO empty interrupt mask 0: masked 1: not masked
[3]	RW	timeoutmsk	Timeout interrupt mask (non-isochronous endpoints) 0: masked 1: not masked
[2]	RW	ahberrmsk	AHB error interrupt mask 0: masked 1: not masked
[1]	RW	epdisbldmsk	Endpoint disable interrupt mask 0: masked 1: not masked
[0]	RW	xfercomplmsk	Transfer completion interrupt mask 0: masked 1: not masked

DOEPEACHMSKN

DOEPEACHMSKN is an interrupt register for device OUT endpoint *n*.



	Offset Address	Register Name	Total Reset Value
	0x0880 + (0x0004 x n) (n = 0–15)	DOEPEACHMSKN	
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
Name	reserved		
Reset	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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 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]	-	reserved	Reserved
[14]	RW	nyetmsk	NYET interrupt mask 0: masked 1: not masked
[13]	RW	nakmsk	NAK interrupt mask 0: masked 1: not masked
[12]	RW	bbleerrmsk	Babble error interrupt mask 0: masked 1: not masked
[11:10]	-	reserved	Reserved
[9]	RW	bnaoutintrmsk	BNA interrupt mask 0: masked 1: not masked
[8]	RW	outpkterrmsk	OUT packet error interrupt mask 0: masked 1: not masked
[7]	-	reserved	Reserved
[6]	RW	back2backsetup	Back-to-back setup packets RX interrupt mask 0: masked 1: not masked
[5]	-	reserved	Reserved
[4]	RW	outtknepdismsk	OUT token RX when endpoint disabled interrupt mask 0: masked 1: not masked



[3]	RW	setupmsk	Setup phase done interrupt mask 0: masked 1: not masked
[2]	RW	ahberrmsk	AHB error interrupt mask 0: masked 1: not masked
[1]	RW	epdisbldmsk	Endpoint disabled interrupt mask 0: masked 1: not masked
[0]	RW	xfercomplmsk	Transfer completion interrupt mask 0: masked 1: not masked

DIEPCTL0

DIEPCTL0 is a control register for device control IN endpoint 0.

Offset Address										Register Name										Total Reset Value												
0x0900										DIEPCTL0										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	epena	epdis	reserved	snak	cnak	txfnum			stall	reserved	eptype	naksts	reserved	usbactep	nextep			reserved					mps									
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		
Bits	Access		Name			Description																										
[31]	RWSC		epena			Endpoint enable 0: disabled 1: transfer completed																										
[30]	RWSC		epdis			Endpoint disable This bit is enabled only in DMA mode.																										
[29:28]	-		reserved			Reserved																										
[27]	WO		snak			NAK setting																										
[26]	WO		cnak			NAK clear																										
[25:22]	RW		txfnum			Number of TX FIFOs																										
[21]	RWSC		stall			Stall handshake This bit is cleared when the endpoint receives a setup token.																										



[20]	-	reserved	Reserved
[19:18]	RO	eptype	Endpoint type
[17]	RO	naksts	NAK status 0: non-NAK handshake TX 1: NAK handshake TX
[16]	-	reserved	Reserved
[15]	RO	usbactep	USB-activated endpoint
[14:11]	-	nextep	Next endpoint This field specifies the ID of the next endpoint that receives data.
[10:2]	-	reserved	Reserved
[1:0]	RW	mps	Minimum packet size 00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes

DOEPCTL0

DOEPCTL0 is a control register for device control OUT endpoint 0.



Offset Address												Register Name												Total Reset Value											
0x0B00												DOEPCTL0												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	epena	epdis	reserved		snak	cnak	reserved				stall	snp	eptype	naksts	reserved	usbactep	reserved								mps										
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					
Bits		Access		Name				Description																											
[31]		RWSC		epena				Endpoint enable When the scatter/gather DMA is enabled: 0: The descriptor structure and the data RX function of the data buffer are disabled. 1: The descriptor structure and the data RX function of the data buffer are enabled. When the scatter/gather DMA is disabled: 0: The function of receiving data from the USB module to the memory is disabled. 1: The function of receiving data from the USB module to the memory is enabled.																											
[30]		RO		epdis				Endpoint disable																											
[29:28]		-		reserved				Reserved																											
[27]		WO		snak				NAK setting																											
[26]		WO		cnak				NAK clear																											
[25:22]		-		reserved				Reserved																											
[21]		RWSC		stall				Stall handshake																											
[20]		RW		snp				Snoop mode																											
[19:18]		RO		eptype				Endpoint type																											
[17]		RO		naksts				NAK status 0: non-NAK handshake TX 1: NAK handshake TX																											
[16]		-		reserved				Reserved																											
[15]		RO		usbactep				USB-activated endpoint																											
[14:2]		-		reserved				Reserved																											



[1:0]	RW	mps	Minimum packet size 00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes
-------	----	-----	--

DIEPCTLN

DIEPCTLN is a control register for device IN endpoint n .

Bit	Offset Address 0x0900 + (0x0020 x n) (n = 0–15)																Register Name DIEPCTLN								Total Reset Value 0x0000_0000							
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EPEna	EPDis	SetD1PID_SetOddFr	SNAK	CNAK	TxFNum				Stall	Sn p	EPType	NAKSis	DPID_EO_FrNum	USBActEP	NextEp				MPS												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bits	Access	Name				Description																										
[31]	RWSC	EPEna				Endpoint enable When the scatter/gather DMA is enabled: 0: The descriptor structure and the data TX function of the data buffer are disabled. 1: The descriptor structure and the data TX function of the data buffer are disabled. When the scatter/gather DMA is disabled: 0: Data on endpoints is not ready. 1: Data on endpoints is ready.																										
[30]	RWSC	EPDis				Endpoint disable																										
[29]	WO	SetD1PID_SetOddFr				DATA1 PID/odd frame																										
[28]	WO	SetD0PID_SetEvenFr				DATA0 PID or even frame in non-scatter/gather DMA mode Scatter/Gather DMA mode: reserved																										
[27]	WO	SNAK				NAK setting																										
[26]	WO	CNAK				NAK clear																										



[25:22]	RW	TxFNum	Number of TX FIFOs Shared FIFO operation: 0: Non-periodic TX FIFO Other values: specified periodic TX FIFO number Dedicated FIFO operation: Number of FIFOs This field is valid only for IN endpoints.
[21]	RW	Stall	Stall handshake (RW) Endpoint control (RWSC)
[20]	RW	Snp	Snoop mode
[19:18]	RW	EPType	Endpoint type 00: control 01: isochronous 10: bulk 11: interrupt
[17]	RO	NAKSts	NAK status 0: non-NAK handshake TX 1: NAK handshake TX
[16]	RO	DPID_EO_FrNum	Endpoint data PID 0: DATA0 1: DATA1 Even/Odd frame Non-scatter/gather DMA mode: 0: even frame 1: odd frame Scatter/Gather DMA mode: reserved
[15]	RWSC	USBActEP	USB-activated endpoint
[14:11]	RW	NextEp	Next endpoint This field specifies the ID of the next endpoint that receives data.
[10:0]	RW	MPS	Maximum packet size

DOEPCTLN

DOEPCTLN is a control register for device OUT endpoint *n*.



Offset Address 0x0B00 + (0x0020 x n) (n = 0–15)												Register Name DOEPCTLN						Total Reset Value 0x0000_0000														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EPEna	EPDis	SetD1PID_SetOddFr	SetD0PID_SetEvenFr	SNAK	CNAK	TxFNum				Stall	Sn p	EPType	NAKSts		DPID_EO_FrNum	USBActEP	NextEp				MPS										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Bits	Access	Name		Description																												
[31]	RWSC	EPEna		Endpoint enable When the scatter/gather DMA is enabled: 0: The descriptor structure and the data RX function of the data buffer are disabled. 1: The descriptor structure and the data RX function of the data buffer are enabled. When the scatter/gather DMA is disabled: 0: The function of receiving data from the USB module to the memory is disabled. 1: The function of receiving data from the USB module to the memory is enabled.																												
[30]	RWSC	EPDis		Endpoint disable																												
[29]	WO	SetD1PID_SetOddFr		DATA1 PID/odd frame																												
[28]	WO	SetD0PID_SetEvenFr		DATA0 PID or even frame in non-scatter/gather DMA mode Scatter/Gather DMA mode: reserved																												
[27]	WO	SNAK		NAK setting																												
[26]	WO	CNAK		NAK clear																												
[25:22]	RW	TxFNum		Number of TX FIFOs Shared FIFO operation: 0: Non-periodic TX FIFO Other values: specified periodic TX FIFO number Dedicated FIFO operation: Number of FIFOs This field is valid only for IN endpoints.																												



[21]	RW	Stall	Stall handshake (RW) Endpoint control (RWSC)
[20]	RW	Snp	Snoop mode
[19:18]	RW	EPType	Endpoint type 00: control 01: isochronous 10: bulk 11: interrupt
[17]	RO	NAKSts	NAK status 0: non-NAK handshake TX 1: NAK handshake TX
[16]	RO	DPID_EO_FrNum	Endpoint data PID 0: DATA0 1: DATA1 Even/Odd frame Non-scatter/gather DMA mode: 0: even frame 1: odd frame Scatter/Gather DMA mode: Reserved
[15]	RWSC	USBActEP	USB-activated endpoint
[14:11]	RW	NextEp	Next endpoint This field specifies the ID of the next endpoint that receives data.
[10:0]	RW	MPS	Maximum packet size

DIEPINTn

DIEPINTn is an interrupt register for device IN endpoint n .



Offset Address 0x0908 + (0x0020 x n) (n = 0–15)																Register Name DIEPINTn								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																StupPktRcvd	NYETIntrpt	NAKIntrpt	BbleErrIntrpt	PktDrpSts	reserved	BNAIntr	TxfifoUndrm_OutPktErr	TxFEmp	INEPNakEff_Back2BackSETUp	INTknEPMis_StsPhaseRevd	INTknTXFEmp_OUTTknEPdis	TimeOUT_SetUp	AHBErr	EPDisbld	XferCompl
Reset	0	0	0	0	0	0	0	0	0	0	0	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	reserved		Reserved																												
[15]	RWSC	StupPktRcvd		Whether setup packets are received 0: no 1: yes																												
[14]	RWSC	NYETIntrpt		NYET interrupt NYET response TX interrupt 0: An interrupt is cleared. 1: An interrupt is generated.																												
[13]	RWSC	NAKIntrpt		NAK interrupt NAK TX interrupt 0: An interrupt is cleared. 1: An interrupt is generated.																												
[12]	RWSC	BbleErrIntrpt		Babble error interrupt Endpoint received babble interrupt. 0: An interrupt is cleared. 1: An interrupt is generated.																												
[11]	RWSC	PktDrpSts		Packet drop status interrupt 0: An interrupt is cleared. 1: An interrupt is generated.																												
[10]	-	reserved		Reserved																												



[9]	RWSC	BNAIntr	BNA interrupt An interrupt is generated when descriptor access is not ready. 0: An interrupt is cleared. 1: An interrupt is generated.
[8]	RWSC	TxfifoUndrn_OutPktErr	FIFO underrun interrupt An interrupt is generated when FIFO underrun occurs. The interrupt is valid when the following conditions are met: -Thresholding enable OUT packet error An interrupt is generated when an overflow or CRC error is detected. The interrupt is valid when the following conditions are met: -Thresholding enable
[7]	RO	TxFEmp	TX FIFO empty status interrupt An interrupt is generated when the TX FIFO is empty or half empty. Note: This bit is valid only for IN endpoints. 0: An interrupt is cleared. 1: An interrupt is generated.
[6]	RWSC	INEPNakEff_Back2BackSETup	IN endpoint NAK effective interrupt An interrupt is generated when the IN endpoint bit is set. Back-to-back setup package received interrupt. An interrupt is generated when the core receives more than three back-to-back setup packets. 0: An interrupt is cleared. 1: An interrupt is generated.
[5]	RWSC	INTknEPMis_StsPhseRcvd	An interrupt is generated when the IN tokens with mismatch EP are received. A control write pulse is received. 0: An interrupt is cleared. 1: A control write pulse is received.
[4]	RWSC	INTknTXFEmp_OUTTknEPdis	An interrupt is generated if IN tokens are received when the corresponding TX FIFO is empty. An interrupt is generated if OUT tokens are received when endpoints are disabled. 0: An interrupt is cleared. 1: An interrupt is generated.



[3]	RWSC	TimeOUT_SetUp	Timeout interrupt An interrupt is generated when a timeout is detected. Setup phase done interrupt. 0: An interrupt is cleared. 1: An interrupt is generated.
[2]	RWSC	AHBErr	AHB error interrupt An interrupt is generated when an AHB error occurs during AHB read/write operations in internal DMA mode. 0: An interrupt is cleared. 1: An interrupt is generated.
[1]	RWSC	EPDisbld	Endpoint disable interrupt 0: An interrupt is cleared. 1: An interrupt is generated.
[0]	-	XferCompl	Transfer completion interrupt 0: An interrupt is cleared. 1: An interrupt is generated.

DOEPINTn

DOEPINTn is an interrupt register for device OUT endpoint *n*.

Offset Address 0x0B08 + (0x0020 x n) (n = 0–15)										Register Name DOEPINTn								Total Reset Value 0x0000_0000														
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved															StupPktRcvd	NYETIntrpt	NAKIntrpt	BleErIntrpt	PltDipSts	reserved	BNALntr	TxfifoUndrv_OutPktErr	TxFEmp	INEPNakEff_Bck2BackSETup	INTknEPMis_SisPhaseRcvd	INTknTXFEmp_OUTTknEPdis	TimeOUT_SetUp	AHBErr	EPDisbld	XferCompl	
Reset	0	0	0	0	0	0	0	0	0	0	0	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	reserved			Reserved																											
[15]	RWSC	StupPktRcvd			Whether setup packets are received 0: no 1: yes																											



[14]	RWSC	NYETIntrpt	NYET interrupt NYET response TX interrupt 0: An interrupt is cleared. 1: An interrupt is generated.
[13]	RWSC	NAKIntrpt	NAK interrupt NAK TX interrupt 0: An interrupt is cleared. 1: An interrupt is generated.
[12]	RWSC	BbleErrIntrpt	Babble error interrupt Endpoint received babble interrupt 0: An interrupt is cleared. 1: An interrupt is generated.
[11]	RWSC	PktDrpSts	Packet drop status interrupt 0: An interrupt is cleared. 1: An interrupt is generated.
[10]	-	reserved	Reserved
[9]	RWSC	BNAIntr	BNA interrupt An interrupt is generated when descriptor access is not ready. 0: An interrupt is cleared. 1: An interrupt is generated.
[8]	RWSC	TxfifoUndrn_OutPktErr	FIFO underrun interrupt An interrupt is generated when FIFO underrun occurs. The interrupt is valid when the following conditions are met: -Thresholding enable OUT packet error An interrupt is generated when an overflow or CRC error is detected. The interrupt is valid when the following conditions are met: -Thresholding enable
[7]	RO	TxFEMp	TX FIFO empty status interrupt An interrupt is generated when the TX FIFO is empty or half empty. This bit is valid only for IN endpoints. 0: An interrupt is cleared. 1: An interrupt is generated.



[6]	RWSC	INEPNakEff_Back2BackSETup	IN endpoint NAK effective interrupt An interrupt is generated when the IN endpoint bit is set. Back-to-back setup package received interrupt An interrupt is generated when the core receives more than three back-to-back setup packets. 0: An interrupt is cleared. 1: An interrupt is generated.
[5]	RWSC	INTknEPMis_StsPhseRcvd	An interrupt is generated when the IN tokens with mismatch EP are received. A control write pulse is received. 0: An interrupt is cleared. 1: An interrupt is generated.
[4]	RWSC	INTknTXFEmp_OUTTknEPdis	An interrupt is generated if IN tokens are received when the corresponding TX FIFO is empty. An interrupt is generated if OUT tokens are received when endpoints are disabled. 0: An interrupt is cleared. 1: An interrupt is generated.
[3]	RWSC	TimeOUT_SetUp	Timeout interrupt An interrupt is generated when a timeout is detected. Setup phase done interrupt 0: An interrupt is cleared. 1: An interrupt is generated.
[2]	RWSC	AHBErr	AHB error interrupt An interrupt is generated when an AHB error occurs during AHB read/write operations in internal DMA mode. 0: An interrupt is cleared. 1: An interrupt is generated.
[1]	RWSC	EPDisblld	Endpoint disable interrupt 0: An interrupt is cleared. 1: An interrupt is generated.
[0]	-	XferCompl	Transfer completion interrupt 0: An interrupt is cleared. 1: An interrupt is generated.

DIEPTSIZ0

DIEPTSIZ0 is a transfer size register for device IN endpoint 0.



DOEPTSIZ0

DOEPTSIZE0 is a transfer size register for device OUT endpoint 0.



DIEPTSIZn

`DIEPTSIZn` is a transfer size register for device IN endpoint n .

DOEPTSIZn

DOEPTSIZn is a transfer size register for device OUT endpoint *n*.



Offset Address 0x0B10 + (0x0020 x n) (n = 0–15)																Register Name DOEPTSIzn								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	mc_rxpid_supcnt	PktCnt												XferSize																	
Reset	0	0	0	0	0	0	0	0	0	0	0	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	reserved		Reserved																												
[30:29]	RW	mc_rxpid_supcnt		Number of packets (IN endpoint, RW) 00: 0 packets 01: 1 packet 10: 2 packets 11: 3 packets Number of packets (non-periodic IN endpoints, RO) This field is valid only in internal DMA mode. PID of the received data (isochronous OUT endpoints, RO) 00: DATA0 01: DATA2 10: DATA1 11: MDATA																												
[28:19]	RW	PktCnt		Number of packets																												
[18:0]	RW	XferSize		Transferred data size																												

DIEPDMAN

DIEPDMAN is a DMA address register for device IN endpoint *n*.



Offset Address																Register Name								Total Reset Value													
0x0914 + (0x0020 x n)																DIEPDMAN								0x0000_0000													
(n = 0–15)																																					
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name																	dmaaddr																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	dmaaddr				DMA address																														

DOEPDMAN

DOEPDMAN is a DMA address register for device OUT endpoint *n*.

Offset Address																Register Name								Total Reset Value													
0x0B14 + (0x0020 x n)																DOEPDMAN								0x0000_0000													
(n = 0–15)																																					
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name																	dmaaddr																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	dmaaddr				DMA address																														

DIEPDMABN

DIEPDMABN is a DMA buffer address register for device IN endpoint *n*.

Offset Address																Register Name								Total Reset Value													
0x091C + (0x0020 x n)																DIEPDMABN								0x0000_0000													
(n = 0–15)																																					
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name																	DMABufferAddr																				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	DMABufferAddr				DMA buffer address																														



DOEPDMABN

DOEPDMABN is a DMA buffer address register for device OUT endpoint n .

DTXFSTS_n

DTXFSTS_n is a TX FIFO status register for device IN endpoints.

Offset Address								Register Name								Total Reset Value																
0x0938								DTXFSTSn								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																INEPTxFSpAvail															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	INEPTxFSpAvail			Available space for the IN endpoint TX FIFO 0x0: The endpoint TX FIFO is full. 0x1: One word is available. 0x2: Two words are available. 0xn: n words ($0 \leq n \leq 32768$) are available. 0x8000: 32768 words are available. Other values: reserved																											

11.8 USB 3.0 Host

11.8.1 Overview

The USB 3.0 module supports the 5 Gbit/s transfer rate defined in the USB 3.0 protocol and 480 Mbit/s transfer rate defined in the USB 2.0 protocol. It complies with the XHCI 1.0

protocol, PHY Interface for the PCI Express (PIPE) protocol (for super-speed transfer), and UTMI protocol (for high-speed transfer). It has an integrated root hub for extending the USB port or other hubs. The USB 3.0 module has the following functions:

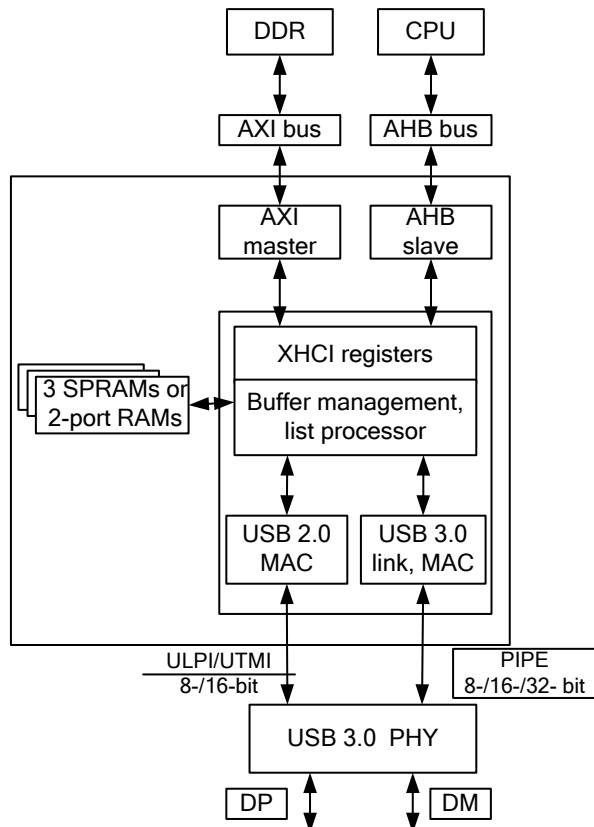
- Controls and processes data transfer.
- Parses data packets and packages data.
- Encodes and decodes the signals transmitted over the USB port.
- Provides interfaces (such as the interrupt vector interface) for the driver.

11.8.2 Function Description

Logic Block Diagram

[Figure 11-44](#) shows the logic block diagram of the USB 3.0 host controller.

Figure 11-44 Logic block diagram of the USB 3.0 host



Typical Applications

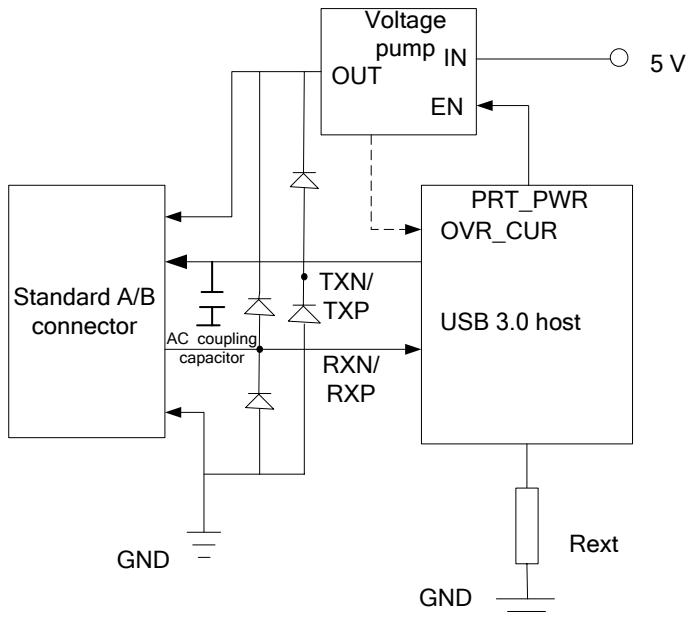
[Figure 11-45](#) shows the reference design of the USB 3.0 host.



CAUTION

- The single-ended impedance of DP or DM is $45\ \Omega \pm 1\%$, and no extra matched resistors are required.
- The precision of the REXT resistor is $\pm 1\%$.
- If high-speed ESD components are used, 1 pF capacitors are recommended.

Figure 11-45 USB 3.0 host reference design



Functions

The USB 3.0 host controller has the following features:

- Complies with the USB 3.0, USB 2.0, USB 1.0, and XHCI 1.0 protocols.
- Independently works in host or device mode.
- Supports super-speed, high-speed, full-speed, and low-speed devices.
- Supports the USB 2.0 low-power solution and USB 3.0 power consumption states (U0, U1, U2, and U3).
- Supports control transfer, bulk transfer, isochronous transfer, and interrupt transfer in host mode.
- Supports control transfer, bulk transfer, and interrupt transfer in device mode.
- Supports the internal DMA controller.
- Supports a maximum of 127 devices by using USB hubs.

Function Implementation

The USB 3.0 host supports the following four standard transfer modes:



- Control transfer

This mode applies to the data transfer between endpoints 0 of the USB host and USB device. For the USB devices of specific models, other endpoints may be used. The control transfer is bidirectional and the transferred data amount is small. Depending on the device and transfer speed, 8-byte data, 16-byte data, 32-byte data, or 64-byte data can be transferred.

- Bulk transfer

This mode is typically used when a large amount of data is transmitted or received with no requirements on the bandwidth and time interval. This mode is the best choice when the transfer speed is very low and many data transfers are delayed. Bulk transfer is performed after all other types of data transfers are complete. In bulk transfer mode, data is transmitted between the USB host and USB device without errors by using an error detection and retransmission mechanism.

- Isochronous transfer

This mode applies to the stream data transfer with strict time requirements and high error tolerance or the instant data transfer at a constant transfer rate. This mode provides a specific bandwidth and time interval.

- Interrupt transfer

This mode applies to transfer of small-sized, scattered, and unpredictable data. In this mode, the device is regularly checked for interrupt data to be sent. The query frequency ranges from 1 ms to 255 ms, which is determined by the device endpoint mode. Typically, the interrupt transfer is unidirectional and only input is available for the USB host.

11.8.3 Operating Mode

Clock Reset

The clock and reset registers must be configured before the USB controller is initialized.

The working clock is disabled as follows:

Step 1 Write 32'h31ff to PERI_CRG44 to reset the USB controller.

Step 2 Write 32'h501 to PERI_CRG45 to configure the PHY.

Step 3 Write 32'h1ff to PERI_CRG44 to deassert the reset on the USB controller, and configure the clock of the USB controller.

----End

11.8.4 Register Variables

Table 11-16 describes the value range and meaning of the variables in the offset addresses for USB 3.0 registers.

Table 11-16 Variables in the offset addresses for USB 3.0 registers

Variable	Value Range
e	0–31
f	0–31



Variable	Value Range
p1	0–15
p2	0–15

11.8.5 Register Summary

Table 11-17 describes USB 3.0 host registers.

Table 11-17 Summary of USB 3.0 host registers (base address: 0xF98A_0000)

Offset Address	Register	Description	Page
0xC100	PERI_USB3_GSBUS_CFG0	Global SoC bus configuration register 0	11-183
0xC104	PERI_USB3_GSBUS_CFG1	Global SoC bus configuration register 1	11-185
0xC108	PERI_USB3_GTXT_HRCFG	Global TX threshold control register	11-186
0xC10C	PERI_USB3_GRXT_HRCFG	Global RX threshold control register	11-186
0xC110	PERI_USB3_GCTL	Global core control register	11-187
0xC118	PERI_USB3_GSTS	Global status register	11-191
0xC11C	PERI_USB3_GUCTL_1	Global user control register 1	11-192
0xC120	PERI_USB3_GSNPS_ID	Global Synopsys ID register	11-193
0xC124	PERI_USB3_GGPIO	Global GPIO register	11-193
0xC128	PERI_USB3_GUID	Global user ID register	11-193
0xC12C	PERI_USB3_GUCTL	Global user control register	11-194
0xC130	PERI_USB3_GBUSE_RRAADDR_LO	Global bus address error register	11-195
0xC134	PERI_USB3_GBUSE_RRAADD_HI	Global bus address error register	11-196
0xC200 + 0x04 x p1	PERI_USB3_GUSB2_PHYCFGN	Global USB 2.0 PHY configuration register	11-196
0xC2C0 + 0x04 x p2	PERI_USB3_GUSB3_PIPECTLN	Global USB 3.0 PIPE control register	11-198
0xC300 + 0x04 x f	PERI_USB3_GTXFI_FOSIZN	Global TX FIFO size register	11-201



Offset Address	Register	Description	Page
0xC380 + 0x04 x f	PERI_USB3_GRXFI_FOSIZN	Global RX FIFO size register	11-202
0xC400 + 0x10 x e	PERI_USB3_GEVN_TADRN_LO	Global event buffer address register	11-202
0xC404 + 0x10 x e	PERI_USB3_GEVN_TADRN_HI	Global event buffer address register	11-202
0xC408 + 0x10 x e	PERI_USB3_GEVN_TSIZN	Global event buffer size register	11-203
0xC40C + 0x10 x e	PERI_USB3_GEVN_TCOUNTN	Global event buffer count register	11-203
0xC610	PERI_USB3_GTXFI_FOPRIDEV	Peripheral global TX FIFO DMA priority register	11-204
0xC618	PERI_USB3_GTXFI_FOPRIHST	Host global TX FIFO DMA priority register	11-204
0xC61C	PERI_USB3_GRXFI_FOPRIHST	Host global RX FIFO DMA priority register	11-205
0xC620	PERI_USB3_GFIFO_PRIDBC	Host global performance debug DMA priority register	11-205
0xC624	PERI_USB3_GDMA_HLRATIO	High/Low priority ratio register for the host global FIFO DMA	11-206
0xC630	PERI_USB3_GFLAD_J	Global frame length adjustment register	11-207
0x0134	PERI_USB5	USB 3.0 system controller 0	11-208
0x0138	PERI_USB6	USB 3.0 PCS system controller 1	11-209
0x013C	PERI_USB7	USB 3.0 PHY system controller 0	11-210
0x0140	PERI_USB8	USB 3.0 PHY system controller 1	11-211
0x0144	PERI_USB9	USB 3.0 PHY system controller 2	11-212
0x0148	PERI_USB10	USB 3.0 PHY system controller 3	11-214
0x014C	PERI_USB12	USB 3.0 system controller 3	11-215

11.8.6 Register Description

PERI_USB3_GSBUSCFG0

PERI_USB3_GSBUSCFG0 is global SoC bus configuration register 0.





[3]	RW	incr16brstena	16-beat burst transfer enable for the AHB master INCR 0: disabled 1: enabled
[2]	RW	incr8brstena	8-beat burst transfer enable for the AHB master INCR 0: disabled 1: enabled
[1]	RW	incr4brstena	4-beat burst transfer enable for the AHB master INCR 0: disabled 1: enabled
[0]	RW	incrbrstena	1-beat burst transfer enable for the AHB master INCR 0: disabled 1: enabled

PERI_USB3_GSBUSCFG1

PERI_USB3_GSBUSCFG1 is global SoC bus configuration register 1.

	Offset Address 0xC104																Register Name PERI_USB3_GSBUSCFG1								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																en1kpage	pipetranslimit		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																												
[31:13]	RW	reserved		Reserved																												
[12]	RW	en1kpage		Boundary select 0: 4 KB boundary 1: 1 KB boundary																												
[11:8]	RW	pipetranslimit		Number of AXI master outstanding requests 0x0: 1 request 0x1: 2 requests 0x2: 3 requests 0x3: 4 requests ... 0xF: 16 requests																												
[7:0]	-	reserved		Reserved																												



PERI_USB3_GTXTHRCFG

PERI_USB3_GTXTHRCFG is a global TX threshold control register.

Bit	Offset Address								Register Name								Total Reset Value															
	0xC108								PERI_USB3_GTXTHRCFG								0x0000_0000															
Name	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved		usbtxpktcntsel	reserved	usbtxpktcnt	usbmaxtxburstsize								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		
Bits	Access	Name		Description																												
[31:30]	-	reserved		Reserved																												
[29]	RW	usbtxpktcntsel		USB TX FIFO threshold select. This field is valid only in super-speed mode. 0: The USB module starts data transfer only after all packets are read to the specified TX FIFO. 1: The USB module starts data transfer only after specified packets are read to the specified TX FIFO.																												
[28]	-	reserved		Reserved																												
[27:24]	RW	usbtxpktcnt		TX FIFO threshold. The value range is 1–15.																												
[23:16]	RW	usbmaxtxburstsize		Maximum TX burst size. This field is valid only for the OUT endpoint during bulk transfer, isochronous transfer, or interrupt transfer at a super speed in host mode. The value range is 1–16.																												
[15:0]	-	reserved		Reserved																												

PERI_USB3_GRXTHRCFG

PERI_USB3_GRXTHRCFG is a global RX threshold control register.



Offset Address								Register Name								Total Reset Value																
0xC10C								PERI_USB3_GRXTHRCFG								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	usbtxpktcntsel	reserved	usbttxpktcnt	usbmaxtxburstsize	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		
Bits		Access		Name				Description																								
[31:30]		-		reserved				Reserved																								
[29]		RW		usbttxpktcntsel				USB RX FIFO threshold select. This field is valid only in super-speed mode. 0: The USB module starts data transfer only after all packets are read to the specified RX FIFO. 1: The USB module starts data transfer only after specified packets are read to the specified RX FIFO.																								
[28]		-		reserved				Reserved																								
[27:24]		RW		usbttxpktcnt				RX FIFO threshold. The value range is 1–15.																								
[23:19]		RW		usbmaxtxburstsize				Reserved																								
[18:0]		-		reserved				Maximum RX burst size. This field is valid only for the IN endpoint during bulk transfer, isochronous transfer, or interrupt transfer at a super speed in host mode. The value range is 1–16.																								

PERI_USB3_GCTL

PERI_USB3_GCTL is a global core control register.



Offset Address												Register Name												Total Reset Value											
0xC110												PERI_USB3_GCTL												0x0000_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	pwrdnscale												masterfiltbypass	bypssetaddr	u2rstecn	firmsdwn	prtcapdir	coresoftreset	softipsync	ul12timerscale	debugattach	ramelkSEL	scaledown	disscramble	u2exit_1fps	gbllibernationen	dsblclkting								
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RW	pwrdnscale	Suspend_clk GCTL[31:19] x 16 k = Suspend_clk Note: 32 kHz < Suspend_clk < 125 MHz																																
[18]	RW	masterfiltbypass	Filtering select 0: Filtering is enabled when DWC_USB3_EN_BUS_FILTERS is 1. 1: Filtering is disabled regardless of the value of DWC_USB3_EN_BUS_FILTERS.																																
[17]	RW	bypssetaddr	SetAddress command in device mode 0: The host sends the SetAddress command to the device. 1: The host does not send the SetAddress command to the device. The device reads the value of DCFG[DevAddress] as the address value. This bit is configured only for simulation.																																
[16]	RW	u2rstecn	Super-speed connection 0: When the super-speed connection fails, the device is in high-speed mode. 1: When the super-speed connection fails, the device waits for more than three cycles. This bit is valid only in device mode.																																



[15:14]	RW	frmscldwn	SOF/USOF/ITP interval in super-speed or high-speed mode 0x3: 15.625 µs 0x2: 31.25 µs 0x1: 62.5 µs 0x1: 125 µs The corresponding value needs to be multiplied by 8 in full-speed mode. Maximum packet size of bulk in or bulk out transfer during simulation in xHCI debug mode 0x0: 1024 bytes 0x1: 512 bytes 0x2: 256 bytes 0x3: 128 bytes
[13:12]	RW	prtcapdir	Port function configuration 01: host 10: device 11: OTG 00: reserved
[11]	RW	coresoftreset	Core soft reset select 0: not reset 1: soft reset Note: When the core is soft-reset, all interrupts excluding the interrupts of the following registers are cleared: -GCTL -GUCTL -GSTS -GSNPSID -GGPIO -GUID -GUSB2PHYCFGn -GUSB3PIPECTLn -DCFG -DCTL -DEVTEN -DSTS
[10]	RW	sofitpsync	0: The first port of the UTMI/ULPI PHY is not suspended no matter whether other SS ports are in the Rx.Detect, SS.Disable, or U3 state. 1: The first port of the UTMI/ULPI PHY is not suspended no matter whether other non-SS ports are suspended. Note: This field is valid only when the USB controller is in host mode.



[9]	RW	u1u2timerscale	U1/U2 timer scale-down enable 0: enabled 1: disabled when GCTL bit[5:4] (ScaleDown) is X1
[8]	RW	debugattach	Debug When this field is set to 1: After the Ru/Stop bit in the DCTL register is set to 1, the SS controller enters the polling link state without detecting the connection of the remote device. The timeout period for link LFPS polling is limited. The timeout period for TS1 polling is limited.
[7:6]	RW	ramclksel	RAM clock select 00: bus clock 01: pipe clock 10: pipe/2 clock 11: reserved Note: In host mode, hardware sets this field to 00 (the ram_clk connects to the bus_clk). Therefore, when the SS port is in P3 state, the PIPE clock is disabled, and the USB 2.0 port does not work.
[5:4]	RW	scaledown	Scale-down timing select In high-speed, full-speed, or low-speed mode: 00: All scale-down timings are disabled, and the actual timings are used for simulation. 01: All scale-down timings excluding the timings related to the following functions are enabled: -Speed enumeration -HNP/SRP -Suspend and resume in host mode 10: Only the scale-down timings related to the suspend and resume functions in device mode are enabled. 11: All scale-down timings are enabled. For the SS mode: 00: All scale-down timings are disabled, and the actual timings are used for simulation. 01: The SS scale-down timings are enabled, including: -The number of TxEq training sequences is decreased to 8. -The LFPS polling burst time is reduced to 100 ns. -The LFPS warm reset receive is reduced to 30 μ s. 10: TxEq training sequences are not transmitted. 11: All scale-down timings are enabled.



[3]	RW	disscramble	Scrambling enable 1: disabled 0: enabled
[2]	RW	u2exit_lfps	U2 state exit signal 0: The link layer considers the 248 ns LFPS signal as an active U2 state exit signal. 1: The link layer waits 8 µs after detecting an active U2 state exit signal.
[1]	RW	gblhibernationen	Global hibernation enable 0: disabled. When the PMU accepts status switching between D0 and D3, the internal status of the core is not retained or restored. 1: enabled
[0]	RW	dsblclkgtng	Internal clock gating enable 0: enabled 1: disabled when the core is in LPM mode Note: This field can be set to 1 after power-on reset.

PERI_USB3_GSTS

PERI_USB3_GSTS is a global status register.

	Offset Address 0xC118										Register Name PERI_USB3_GSTS										Total Reset Value 0x0000_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	cbelt										reserved										otg_ip	bc_ip	adp_ip	host_ip	device_ip	crsttimeout	buserraddrvid	reserved	reserved	curmod		
Reset	0	0	0	0	0	0	0	0	0	0	0	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	cbelt		Minimum value among all received device BELT values and the BELT value configured by the Set Latency Tolerance Value command in host mode																												
[19:11]	-	reserved		Reserved																												
[10]	RO	otg_ip		An OTG-related interrupt in the OEVT register is waiting for handling.																												
[9]	RO	bc_ip		A BC-related interrupt in the BCEVT register is waiting for handling.																												



[8]	RO	adp_ip	An ADP-related interrupt in the ADPEVT register is waiting for handling.
[7]	RO	host_ip	An xHCI-related interrupt in the host event queue is waiting for handling.
[6]	RO	device_ip	An xHCI-related interrupt in the device event queue is waiting for handling.
[5]	RO	csrttimeout	The duration of accessing registers by software exceeds the time defined by DWC_USB3_CSR_ACCESS_TIMEOUT.
[4]	RO	buserraddrvld	GBUSERRADDR register validity indicator and start address for the error position.
[3:2]	-	reserved	Reserved
[1:0]	RO	curmod	Current working mode 00: device mode 01: host mode

PERI_USB3_GUCTL1

PERI USB3 GUCTL1 is global user control register 1.



PERI_USB3_GSNPSID

PERI_USB3_GSNPSID is a Synopsys ID register.

Offset Address								Register Name								Total Reset Value																
0xC120								PERI_USB3_GSNPSID								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	synopsysid																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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	synopsysid			The upper 16 bits indicate USB 3.0, and the lower 16 bits indicate 2.50a.																											

PERI_USB3_GGPIO

PERI_USB3_GGPIO is a global GPIO register.

Offset Address								Register Name								Total Reset Value																
0xC124								PERI_USB3_GGPIO								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	gpo												gpi																			
Reset	0	0	0	0	0	0	0	0	0	0	0	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	gpo			Drive value of gp_o[15:0]																											
[15:0]	RO	gpi			Read value of gp_i[15:0]																											

PERI USB3 GUID

PERI USB3 GUID is a global user ID register.

Offset Address								Register Name								Total Reset Value																
0xC128								PERI_USB3_GUID								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	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		
Bits	Access	Name				Description																										
[31:0]	RO	reserved				User information including the system version and hardware configuration																										



PERI_USB3_GUCL

PERI_USB3_GUCTL is a global user control register.



[14]	RW	usbhstlnautoretryen	Automatic retransmission enable for host inputs 0: disabled If an error occurs during host input transfer, the host automatically replies an ACK indicating termination to the device (Retry = 1 and NumP = 0). 1: enabled If automatic retransmission is enabled and an error occurs during host input transfer, the host automatically replies an ACK not indicating termination to the device (Retry = 1 and NumP != 0).
[13]	RW	enoverlapchk	LFPS overlap signal detection enable 1: The LFPS overlap signal is detected to avoid glitches. 0: The LFPS overlap signal is not detected.
[12]	RW	extcapsupten	External additional function enable 1: enabled 0: disabled
[11]	RW	csr	Extra delay inserted between full-speed BULKOUT transfers
[10:9]	RW	dtct	Rough timeout period that the device responds to the host. If this field is set to 0, the timeout period is defined by the DTFT; if this field is not 0, the timeout period is as follows: 01: 500 µs 10: 1.5 ms 11: 6.5 ms
[8:0]	RW	dtft	Exact timeout period that the device responds to the host. This field is valid when the DTCT is 0. $T = DTFT \times 256 \times 8 \mu s$

PERI_USB3_GBUSERRADDR_LO

PERI_USB3_GBUSERRADDR_LO is a global bus address error register.

	Offset Address																Register Name								Total Reset Value							
	0xC130																PERI_USB3_GBUSERRADDR_LO								0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	busaddrlo																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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	busaddrlo		Lower 32 bits of the error address This field is valid only when GSTS[BusErrAddrVld] is 1. This field is cleared only during reset, and it supports only the AHB and AXI bus configurations.																												



PERI_USB3_GBUSERRADD_HI

PERI_USB3_GBUSERRADD_HI is a global bus address error register.

	Offset Address																Register Name																Total Reset Value							
	0xC134																PERI_USB3_GBUSERRADD_HI																0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	busaddrhi																																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	busaddrhi			Upper 32 bits of the error address This field is valid only when GSTS[BusErrAddrVld] is 1. This field is cleared only during reset, and it supports only the AHB and AXI bus configurations.																																			

PERI_USB3_GUSB2PHYCFGN

PERI_USB3_GUSB2PHYCFGN is a global USB 2.0 PHY configuration register.

	Offset Address																Register Name																Total Reset Value											
	0xC200 + 0x04 x p1																PERI_USB3_GUSB2PHYCFGN																0x0000_0000											
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
Name	phy_soft_rst <ul style="list-style-type: none">phy_freeclk_existsulpi_lpnm_with_opmode_chk reserved <ul style="list-style-type: none">ulpi_ext_vbus_indicatorulpi_ext_vbus_drvreservedulpi_auto_resreserved usbtrdtim <ul style="list-style-type: none">xcvrdlyenblslpmphyselsusphyfsintfulpi_utmi_selphyiftoutcal																																											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	phy_soft_rst			UTMI PHY reset by triggering the usb2phy_reset signal In this case, the ULPI PHY is not reset, because it is reset by configuring the FunctionControl[Reset] register. When the core is reset, the core automatically configures vcc_reset_n, USBCMD[HCRST], DCTL[SoftReset], or GCTL[SoftReset] to implement reset.																																							



[30]	RW	u2_freeclk_exists	Whether the USB 2.0 PHY provides the free_clk 0: no 1: yes
[29]	RW	ulp1_lpm_with_op_mode_chk	NOPID TX enable of the ULPI PHY in LPM mode 0: In LPM mode, the controller transmits NOPID before transmitting EXTPID. 1: In LPM mode, the controller does not transmit NOPID before transmitting EXTPID.
[28:19]	RW	reserved	Reserved
[18]	RW	ulp1_ext_vbus_indicator	ULPI external Vbus indicator 0: The PHY uses the internal Vbus valid comparator. 1: The PHY uses the external Vbus valid comparator.
[17]	RW	ulp1_ext_vbus_drv	ULPI external Vbus indicator 0: The PHY uses the internal Vbus valid comparator. 1: The PHY uses the external Vbus valid comparator.
[16]	RW	reserved	Reserved
[15]	RW	ulp1_auto_res	ULPI automatic resume enable 0: disabled 1: enabled
[14]	RW	reserved	Reserved
[13:10]	RW	usbtrdtim	USB 2 turnaround time, that is, response time after the MAC requests the packet FIFO controller (PFC) to fetch data from the DFIFO (SPRAM) 0x5 for the 16-bit UTMI+ interface 0x9 for the 8-bit UTMI+ or ULPI interface
[9]	RW	xcvrdly	TX/RX delay. When this field is set to 1, the 2.5 μ s delay is added between the time when Transceiver Select is set to 00 (high speed) and the time when TxValid is 0 to transmit the chirp-K handshake signal.
[8]	RW	enblslpm	Whether the utmi_sleep_n and utmi_11_suspend_n signals connect to the PHY 0: The utmi_sleep_n and utmi_11_suspend_n signals do not connect to the PHY. 1: The utmi_sleep_n and utmi_11_suspend_n signals connect to the PHY.
[7]	RW	physel	PHY interface type 0: USB 2.0 high-speed UTMI+ or ULPI PHY interface 1: USB 1.1 full-speed serial interface This bit is set to 1 when it is write-only.



[6]	RW	susphy	USB 2.0 high-speed/full-speed/low-speed PHY suspend 0: not suspended 1: suspended Note: In host mode, set this field to 1 after the core is initialized.
[5]	RW	fsintf	Type of the full-speed PHY interface 0: 6-pin unidirectional full-speed serial transfer interface 1: 3-pin bidirectional full-speed serial transfer interface When this bit is read-only, the return value is 0.
[4]	RW	ulpi_utmi_sel	Type of the high-speed PHY interface 0: UTMI+ 1: ULPI
[3]	RW	phyif	Data width of the UTMI interface 0: 8 bits 1: 16 bits
[2:0]	RW	toutcal	High-speed/Full-speed timeout calibration The corresponding bit time is added to each PHY clock. High-speed mode: One 30 MHz PHY clock = 16 bit times One 60 MHz PHY clock = 8 bit times Full-speed mode: One 30 MHz PHY clock = 0.4 bit times One 60 MHz PHY clock = 0.2 bit times One 48 MHz PHY clock = 0.25 bit times

PERI_USB3_GUSB3PIPECTLN

PERI_USB3_GUSB3PIPECTLN is a global USB 3.0 PIPE control register.



Offset Address 0xC2C0 + 0x04 x p2 (p2 = 0-15)																Register Name PERI_USB3_GUSB3PIPECTLN								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	phy_soft_rst	hstprtcmpl	u2ssinactp3ok	disrxdetp3	ux_exit_in_px	ping_enhancement_en	u1u2exitfail_to_recov	request_p1p2p3	reserved	reserved	reserved	delaylp2p3	delay phy powerchange	suspend_en	datwidth	abortrxdefini2	skiprxdet	lfps_p0_align	p3p2_tran_ok	p3exsigp2	lfps_filter	polling_lfps_control	reserved	txswing	txmargin	txdeemphasis	elastic_buffer_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		
Bits	Access	Name		Description																												
[31]	RW	phy_soft_rst		USB 3.0 soft reset																												
[30]	RW	hstprtcmpl		PIPE PHY template test without the USB 3.0 cable																												
[29]	RW	u2ssinactp3ok		PHY state select in U2/SSInactive state 0: The PHY enters the P2 state. 1: The PHY enters the P3 state.																												
[28]	RW	disrxdetp3		RX detection enable in P3 state 0: If the PHY is in P3 state and the core requires RX detection, the core performs RX detection in P3 state. 1: If the PHY is in P3 state and the core requires RX detection, the core switches the PHY state to P2 and then performs RX detection. After detection is complete, the core switches the PHY state to P3.																												
[27]	RW	ux_exit_in_px		PHY state select when the core state is switched 0: The PHY is in P0 state when the core exits the U1, U2, or U3 state. 0: The PHY is in P1, P2, or P3 state when the core exits the U1, U2, or U3 state respectively. Note: Set this field to 0 when the Synopsys PHY is used.																												
[26]	RW	ping_enhancement_en		Timeout period changed from 500 ms to 300 ms for the ping command of the downlink port U1 Note: Set this field to 0 when the Synopsys PHY is used.																												
[25]	RW	u1u2exitfail_to_recov		When this bit is set to 1, and U1/U2 LFPS handshake fails, the LTSSM transitions from the U1/U2 state to Recovery state instead of SS Inactive state. If Recovery fails, then the LTSSM can enter the SS Inactive state. This is an enhancement only. It prevents interoperability issues if the remote link does not do proper handshake.																												



[24]	RW	request_p1p2p3	When the core state is switched from U0 to U1, U2, or U3, the core always requests the PHY to switch the state from P0 to P1, P2, or P3 respectively. Note: Set this field to 1 when the Synopsys PHY is used.
[23]	RW	reserved	Reserved
[22]	RW	reserved	Reserved
[21:19]	RW	delayp1p2p3	Delay when the state is switched from P0 to P1, P2, or P3 When the core enters the U1, U2, or U3 status, the time when the P0 state is switched to P1, P2, or P3 is prolonged until Pipe3_RxValid is set to 0 or an 8B10B error occurs. Note: This function is valid only when bit 18 is set to 1.
[18]	RW	delay_phy_powerchange	PHY status switch delay 0: When the core state is switched from U0 to U1, U2, or U3, the time when the PHY enters the P1, P2, or P3 state is delayed until the values of Pipe3_RxElecIdle and pipe3_RxValid are 0. 0: When the core state is switched from U0 to U1, U2, or U3, the PHY enters the P1, P2, or P3 state without querying the values of Pipe3_RxElecIdle and pipe3_RxValid. Note: Set this field to 1 when the Synopsys PHY is used.
[17]	RW	suspend_en	USB 3.0 PHY suspend enable 0: not suspended 1: suspended Note: In host mode, set this field to 1 after the core is initialized.
[16:15]	RW	datwidth	Data width of the PIPE interface 00: 32 bits 01: 16 bits 10: 8 bits 11: reserved
[14]	RW	abortrxdetinu2	RX detection abort in U2 state When this field is set to 1 and the connection state is U2, the core receives a U2 state exit signal from a remote device and RX detection is not performed. Note: Set this field to 0 when the Synopsys PHY is used.
[13]	RW	skiprxdet	RX detection skip. When this field is set to 1, RX detection is skipped if pipe3_RxElecIdle is pulled down.
[12]	RW	lfps_p0_align	When the controller exits the U1, U2, or U3 state, LFPS signal transfer is stopped at the clock edge of the PHY P0 state request signal. Otherwise, the LFPS signal is transmitted one cycle earlier. When the PHY state is switched from P1 or P2 to P0, the controller requests data transfer two clock cycles later after the PHY sets PhyStatus.



[11]	RW	p3p2_tran_ok	P2/P3 state switch 1: The PHY is switched from P2 to P3 or from P3 to P2 without entering the intermediate status P0. 0: The PHY enters the intermediate status P0 each time when the PHY status is switched between P2 and P3. Note: Set this field to 0 when the Synopsys PHY is used.
[10]	RW	p3exsigp2	P3 exit status select 1: When the core exits the U3 state, the PHY state must be switched from P3 to P2. 0: When the core exits the U3 state, the PHY state can be switched from P3 to other state. Note: Set this field to 0 when the Synopsys PHY is used.
[9]	RW	lfps_filter	LFPS filtering signal 1: The controller filters the LFPS signal from the PHY until pipe3_Rxelecidle and pipe3_RxValid are deasserted. 0: The controller does not filter the LFPS signal from the PHY.
[8]	RW	polling_lfps_control	LFPS polling control after RX detection 0: (default) LFPS polling starts 400 µs later after RX detection. 1: LFPS polling starts after RX detection without delay.
[7]	-	reserved	Reserved
[6]	RW	txswing	TX swing of the PIPE interface
[5:3]	RW	txmargin	TX margin of the PIPE interface
[2:1]	RW	txdeemphasis	TX preemphasis of the PIPE interface
[0]	RW	elastic_buffer_mode	Elastic buffer mode

PERI_USB3_GTXFIFOSIZN

PERI_USB3_GTXFIFOSIZN is a global TX FIFO size register.

Offset Address		Register Name																Total Reset Value									
0xC300 + 0x04 x f		PERI_USB3_GTXFIFOSIZN																0x0000_0000									
(f = 0~31)																											
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																										
Name	txfstaddr_n																txfdep_n										
Reset	0 0 0 0 0 0 0 0																										
Bits	Access	Name		Description																							
[31:16]	RW	txfstaddr_n		Start address for the TX FIFO n RAM in the memory																							



[15:0]	RW	txfdep_n	TX FIFO depth Minimum value: 32 MDWIDTH-bit words Maximum value: 32768 MDWIDTH-bit words
--------	----	----------	--

PERI_USB3_GRXFIFOSIZN

PERI_USB3_GRXFIFOSIZN is a global RX FIFO size register.

	Offset Address 0xC380 + 0x04 x f (f = 0-31)																Register Name PERI_USB3_GRXFIFOSIZN								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	rxfstaddr_n																rxfdep_n																
Reset	0	0	0	0	0	0	0	0	0	0	0	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	rxfstaddr_n																Start address for the RX FIFO n RAM in the memory															
[15:0]	RW	rxfdep_n																RX FIFO depth Minimum value: 32 MDWIDTH-bit words Maximum value: 32768 MDWIDTH-bit words															

PERI_USB3_GEVNTADRN_LO

PERI_USB3_GEVNTADRN_LO is a global event buffer address register.

	Offset Address 0xC400 + 0x10 x e (e = 0-31)																Register Name PERI_USB3_GEVNTADRN_LO								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	evntadrlo																																
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RWSC	evntadrlo																Lower 32-bit address of the event buffer															

PERI_USB3_GEVNTADRN_HI

PERI_USB3_GEVNTADRN_HI is a global event buffer address register.



Offset Address 0xC404 + 0x10 x e (e = 0–31)																Register Name PERI_USB3_GEVNTADRN_HI								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	evntadrhi																															
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RWSC	evntadrhi		Upper 32-bit address of the event buffer																												

PERI_USB3_GEVNTSIZN

PERI_USB3_GEVNTSIZN is a global event buffer size register.

Offset Address 0xC408 + 0x10 x e (e = 0–31)																Register Name PERI_USB3_GEVNTSIZN								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	evntintmask												reserved												evntsiz							
Reset	0	0	0	0	0	0	0	0	0	0	0	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	evntintmask		Event interrupt mask																												
[30:16]	-	reserved		Reserved																												
[15:0]	RW	evntsiz		Event buffer size (byte)																												

PERI_USB3_GEVNTCOUNTN

PERI_USB3_GEVNTCOUNTN is a global event buffer count register.



Offset Address 0xC40C + 0x10 x e (e = 0–31)																Register Name PERI_USB3_GEVNTCOUNTN								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																evntcount															
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RWSC	evntcount		When this register is read, the number of valid events in the event buffer is returned; when it is written, the hardware automatically decreases the count value.																												

PERI_USB3_GTXFIFOPRIDEV

PERI_USB3_GTXFIFOPRIDEV is a peripheral global TX FIFO DMA priority register.

Offset Address 0xC610																Register Name PERI_USB3_GTXFIFOPRIDEV								Total Reset Value 0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																device_txfifo_priority															
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved		Reserved																												
[3:0]	0x0	device_txfifo_priority		Priority of the device TX FIFO 1: high 0: low																												

PERI_USB3_GTXFIFOPRIHST

PERI_USB3_GTXFIFOPRIHST is a host global TX FIFO DMA priority register.



Offset Address																Register Name								Total Reset Value								
0xC618																PERI_USB3_GTXFIFOPRIHST								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									host_txfifo_priority						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved			Reserved																											
[3:0]	RW	host_txfifo_priority			Priority of the host TX FIFO 1: high 0: low																											

PERI_USB3_GRXFIFOPRIHST

PERI_USB3_GRXFIFOPRIHST is a host global RX FIFO DMA priority register.

Offset Address																Register Name								Total Reset Value								
0xC61C																PERI_USB3_GRXFIFOPRIHST								0x0000_0000								
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved																									host_rxfifo_priority						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	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]	-	reserved			Reserved																											
[3:0]	RW	host_rxfifo_priority			Priority of the host RX FIFO 1: high 0: low																											

PERI_USB3_GFIFOPRIDBC

PERI_USB3_GFIFOPRIDBC is a host global performance debug DMA priority register.



PERI USB3 GDMAHLRATIO

PERI_USB3_GDMAHLRATIO is a high/low priority ratio register for the host global FIFO DMA.

Offset Address								Register Name								Total Reset Value																												
0xC624								PERI_USB3_GDMAHLRATIO								0x0000_0000																												
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
Name	reserved												hst_rx_fifo_dma_hilo_priority_ratio				reserved				hst_tx_fifo_dma_hilo_priority_ratio																							
Reset	0	0	0	0	0	0	0	0	0	0	0	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																																								
[12:8]	RW	hst_rx_fifo_dma_hilo_priority_ratio		High/Low priority ratio of the host RX FIFO DMA																																								
[7:5]	-	reserved		Reserved																																								



[4:0]	RW	hst_txfifo_dma_hilo_priority_ratio	High/Low priority ratio of the host TX FIFO DMA
-------	----	------------------------------------	---

PERI_USB3_GFLADJ

PERI_USB3_GFLADJ is a global frame length adjustment register.

	Offset Address	Register Name	Total Reset Value
Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	PERI_USB3_GFLADJ	0x0000_0000
Name	gfladj_refclk_240mhzdecr_pls1 gfladj_refclk_240mhz_decr	gfladj_refclk_lpm_sel reserved	gfladj_refclk_fladj gfladj_30mhz_reg_sel 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	gfladj_refclk_240mhzdecr_pls1	Precision of GFLADJ_REFCLK_240MHZ_DECR/ref_frequency 0: The remainder is less than 0.5. 1: The remainder is greater than or equal to 0.5.
[30:24]	RW	gfladj_refclk_240mhz_decr	GFLADJ_REFCLK_240MHZ_DECR = 240/ref_clk_frequency
[23]	RW	gfladj_refclk_lpm_sel	SOF/ITP count clock select If this bit is set to 1, ref_clk is the count clock.
[22]	-	reserved	Reserved
[21:8]	RW	gfladj_refclk_fladj	SOF/ITP calibration value when bit 23 is 1 $\text{FLADJ_REF_CLK_FLADJ} = ((125000/\text{ref_clk_period_integer}) - (125000/\text{ref_clk_period})) \times \text{ref_clk_period}$
[7]	RW	gfladj_30mhz_reg_sel	SOF/ITP calibration select 1: The controller calibrates the SOF/ITP based on the value of PERI_USB3_GFLADJ[gfladj_30mhz]. 0: The controller calibrates the SOF/ITP based on the value of fladj_30mhz_reg.
[6]	-	reserved	Reserved



[5:0]	RW	gfladj_30mhz	When bit 7 is 1 and the SOF/ITP count clock is the UTMI/ULPI clock, the controller calibrates the SOF/ITP based on this field value.
-------	----	--------------	--

PERI_USB5

PERI_USB5 is USB 3.0 system controller 0.

	Offset Address	Register Name	Total Reset Value
Bit	31 30 29 28 27 26 25 24 23 22 21 20	PERI_USB5	0x7E80_2040
Name	host_current_belt	bus_filter_bypass chirp_on reserved	fladj_30mhz_reg host_msi_enable host_port_power_control_present hub_u3_port_disable hub_u2_port_disable hub_port_perm_attach reserved
Reset	0 1 1 1 1 1 1 0 1 0 0 0	0 0 0 0 0 0 1 0 0 0 0 0 0 1 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
[31:20]	RW	host_current_belt	Current BELT value
[19:16]	RW	bus_filter_bypass	UTMI bus signal filtering enable. The value 1 indicates disabled, and the value 0 indicates enabled. bus_filter_bypass[3] The function of filtering out the utmiotg_iddig signal is disabled. bus_filter_bypass[2] The function of filtering out utmisrp_bvalid and utmisrp_sessend signals is disabled. bus_filter_bypass[1] The function of filtering out all pipe3_PowerPresent signal of port U3 is disabled. bus_filter_bypass[0] The function of filtering out all utmiotg_vbusvalid signal of port U2 is disabled.
[15]	RW	chirp_on	Chirp signal detected
[14]	RW	reserved	Reserved



[13:8]	RW	fladj_30mhz_reg	High-speed signal jitter adjust The mac3_clock and utmi_clock are adjusted to 125 μs.
[7]	RW	host_msi_enable	Pulse interrupt enable 1: Interrupt signals are output as pulses. 0: Interrupt signals are output as levels.
[6]	RW	host_port_power_control_present	Port power switch enable 0: The port has no power switch. 1: The port has a power switch.
[5]	RW	hub_u3_port_disable	USB 3.0 super-speed port enable 0: enabled 1: disabled
[4]	RW	hub_u2_port_disable	USB 3.0 high-speed port enable. 0: enabled 1: disabled
[3:2]	RW	hub_port_perm_attach	Device permanent insertion 0: yes 1: no
[1:0]	RW	reserved	Reserved

PERI_USB6

PERI_USB6 is USB 3.0 PCS system controller 1.

	Offset Address																Register Name								Total Reset Value									
	0x0138																PERI_USB6								0x5D81_560D									
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	reserved																pcs_tx_swing_full	pcs_tx_deemh_6db	pcs_tx_deemph_3p5db	lane0_tx2rx_loophk	lane0_power_present_ovrd_en	lane0_power_present_ovrd	lane0_ext_pclk_req	reserved			usb_pwr_ctrl							
Reset	0	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	0	0	0	0	1	1	0	1		
Bits	Access	Name				Description																												
[31]	RW	reserved				Reserved																												



[30:24]	RW	pcs_tx_swing_full	TX signal swing (for the eye pattern test)
[23:18]	RW	pcs_tx_deemh_6db	Preemphasis of 6 dB reduction (for the eye pattern test)
[17:12]	RW	pcs_tx_deemph_3p5db	Preemphasis of 3.5 dB reduction (for the eye pattern test)
[11]	RW	lane0_tx2rx_lopbk	Lane 0 TX-to-RX loopback enable
[10]	RW	lane0_power_preset_ovrd_en	Lane 0 overcurrent protection enable 1: enabled 0: disabled
[9]	RW	lane0_power_preset_ovrd	Lane 0 overcurrent protection 1: enabled 0: disabled
[8]	RW	lane0_ext_pclk_req	Lane 0 external PIPE clock enable 1: enabled 0: disabled
[7:4]	RW	reserved	Reserved
[3:0]	RW	usb_pwr_ctrl	Power control Bit[0]: power switch control signal 1: on 0: off Bit[1]: overcurrent protection enable signal 1: enabled 0: disabled Bit[2]: power switch polarity inversion enable signal 1: inverted 0: not inverted Bit[3]: overcurrent protection polarity inversion enable signal 1: inverted 0: not inverted

PERI_USB7

PERI_USB7 is USB 3.0 PHY system controller 0.



Offset Address								Register Name								Total Reset Value																
0x013C								PERI_USB7								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	crdatain												crdataout																			
Reset	0	0	0	0	0	0	0	0	0	0	0	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	crdatain			Current input data																											
[15:0]	RW	crdataout			Current output data																											

PERI_USB8

PERI_USB8 is USB 3.0 PHY system controller 1.

Offset Address								Register Name								Total Reset Value																
0x0140								PERI_USB8								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	otgtune	otgdisable0	mpllrefssclken	mpllmultiplier	loslevel	losbias	loopbackenb0	idpullup0	iddig0	drvbus0	compdistune0	common0	crack	crwrit	ciread	ercapdata	ercapaddr															
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RW	otgtune		Vbus valid threshold adjust 1 bit = 1.5%																												
[28]	RW	otgdisable0		PHY OTG disable																												
[27]	RW	mpllrefssclken		Spread reference clock output. It is 20 MHz and provides the clock for the hardcore of the clock source of ref_alt_clk.																												
[26:20]	RW	mpllmultiplier		MPLL frequency multiplier control for obtaining a specified working frequency																												
[19:16]	RW	loslevel		Sensitivity level detection for the loss-of-signal (LOS) detector																												
[15:13]	RW	losbias		Level threshold detection for the LOS detector																												
[12]	RW	loopbackenb0		Loopback enable 1: enabled 0: disabled																												



[11]	RW	idpullup0	Sampling enable for the analog ID input signal 1: enabled 0: disabled
[10]	RW	iddig0	Mini-A or mini-B connector connected to the PHY 1: mini-B connector 0: mini-A connector
[9]	RW	drvbus0	Vbus valid comparator enable 1: enabled 0: disabled
[8:6]	RW	compdistune0	Disconnection event detection voltage threshold. This field is used to set the voltage threshold for the disconnection event between the device and the host.
[5]	RW	commomon0	Common module enable 1: The HS bias and PLL modules are disabled in suspend or sleep mode. 0: The HS bias and PLL modules are enabled in suspend or sleep mode.
[4]	RW	crack	Controller signal (as the response to the CRWRITE, CRREAD, CRCAPDATA, and CRCAPSDDR signals)
[3]	RW	crwrite	Register write control signal
[2]	RW	crread	Register read control signal
[1]	RW	crcapdata	Register data capture control signal. cr_data_in[15:0] are transferred to the written data.
[0]	RW	crcapaddr	Register address capture control signal. cr_data_in[15:0] are transferred to the address register.

PERI_USB9

PERI_USB9 is USB 3.0 PHY system controller 2.



	Offset Address 0x0144																Register Name PERI_USB9								Total Reset Value 0x0000_0000							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	txpreemppulse0	txpreempamp tune0	txfusltune0				sscrefclk sel								reserved	txhsxvtune0	tx0termoffset				ssrange	sscen	sqrxtune0				refenable					
Reset	0	0	0	0	0	0	0	0	0	0	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	txpreemppulse0		TX preemphasis time in high-speed mode 1: 580 μs 0: 2 x 580 μs																												
[30:29]	RW	txpreempamp tune0		TX preemphasis tune in high-speed mode 11: 1800 μA 10: 1200 μA 01: 600 00: Preemphasis is disabled																												
[28:25]	RW	txfusltune0		Source impedance tune in full-speed or low-speed mode by using hot codes. The impedance is decreased by 2.5% each time a hot code is added, and vice versa.																												
[24:16]	RW	sscrefclk sel		Spread spectrum reference clock select																												
[15]	RW	reserved		Reserved																												
[14:13]	RW	txhsxvtune0		TX DP/DM voltage tune in high-speed mode 11: default configuration 10: +15 mV 01: -15 mV 00: reserved																												
[12:8]	RW	tx0termoffset		TX termination compensation tune enable 1: enabled 0: disabled																												
[7:5]	RW	ssrange		Spread spectrum clock range																												
[4]	RW	sscen		Spread spectrum enable 1: enabled 0: disabled																												



[3:1]	RW	sqrxtune0	High-speed data detection level tune The level is decreased by 5% each time a binary value is added, and vice versa.
[0]	RW	retenablen	Low digital power indicator, indicating that the VP digital power is decreased in suspend mode. 1: The normal operating mode is used. 0: The analog power is shut down.

PERI_USB10

PERI_USB10 is USB 3.0 PHY system controller 3.

Offset Address								Register Name								Total Reset Value																
0x0148								PERI_USB10								0x0000_0000																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								vbusvldextsel0	vbusvldext0	txvreftune0				reserved	testpowerdownssp		testpowerdownssp		rtunek	rtunreq	vdatsrcenb0	vdalddetenb0	txrisetune0		txrestune0						
Reset	0	0	0	0	0	0	0	0	0	0	0	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]	RW	reserved		Reserved																												
[18]	RW	vbusvldextsel0		External Vbus valid indicator 1: VBUSVLDEXT0 0: internal comparator																												
[17]	RW	vbusvldext0		External Vbus valid indicator signal 1: The Vbus0 signal and the pull-up resistor on D+ are valid. 0: The Vbus0 signal is invalid, and the pull-up resistor on D+ is disabled.																												
[16:13]	RW	txvreftune0		DC voltage tune in high-speed mode The voltage is increased by 1.25% each time the binary value 1 is added, and vice versa.																												
[12:10]	RW	reserved		Reserved																												
[9]	RW	testpowerdownssp		Power-off control for the super-speed function circuit 1: turned off 0: not turned off																												



[8]	RW	testpowerdownhsp	Power-off control for the high-speed function circuit 1: turned off 0: not turned off
[7]	RW	rtunreak	Impedance tune acknowledge enable
[6]	RW	rtunreq	Impedance tune request
[5]	RW	vdatsrcenb0	Charging power select 1: The data source voltage (VDAT_SRC) is enabled. 0: The data source voltage (VDAT_SRC) is disabled.
[4]	RW	vdatdetenb0	Charging connection/disconnection detection enable 1: The data detection voltage is enabled. 0: The data detection voltage is disabled.
[3:2]	RW	txrisetune0	Rising/Falling edge time tune for the TX end in high-speed mode The rising/falling edge time is decreased by 4% each time the binary value 1 is added, and vice versa.
[1:0]	RW	txrestune0	USB matched source impedance tune

PERI_USB12

PERI_USB12 is USB 3.0 system controller 3.

	Offset Address																Register Name								Total Reset Value															
	0x014C																PERI_USB12								0x0000_0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	reserved																								pcs_rx_los_mask_val															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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	reserved			Reserved																																			
[9:0]	RW	pcs_rx_los_mask_val			Number of reference clock cycles for masking the LFPS signals that are being received Value = 10 µs/Tref_clk If ref_clkdiv2 is used, Value = 10 µs/(2 x Tref_clk).																																			



11.9 SCI

11.9.1 Overview

The SCI is located on the APB and is provided for the external card reader. The CPU reads data from or writes data to the smart card through the SCI and implements serial-to-parallel conversion (when reads data from the smart card) and parallel-to-serial conversion (when writes data to the smart card).

11.9.2 Features

The SCI has the following features:

- Supports the ISO/IEC 7816-3, 7816-10, and EMV standards and the T0, T1, and T14 asynchronous transmission protocols.
- Supports the configurable card clock rate and elementary time unit (ETU).
- Supports data transfer without transmitting a parity bit.
- Provides a 32-byte internal TX FIFO and a 32-byte internal RX FIFO.
- Allows hardware to detect the FIFO depth and reports corresponding interrupts.
- Queries 15 types of maskable and independent interrupts and reports one combined interrupt.
- Queries raw and masked interrupt status.
- Allows software to activate a smart card after it is inserted.
- Allows software to release a smart card after it completes data transfer.
- Allows hardware to release a smart card when its removal is detected.
- Supports the SCI soft reset.
- Supports the clock rate conversion factor F of 372 or 512 and the bit rate adjustment factor D of 1, 2, 4, 8, or 16.

11.9.3 Function Description

11.9.3.1 Application Block Diagram

Figure 11-46 shows the typical application circuit of the SCI.

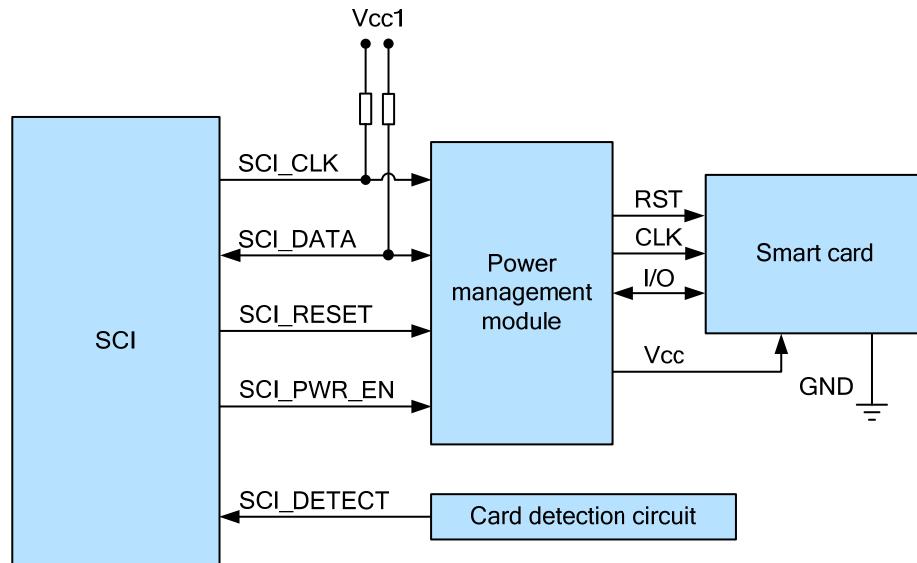


CAUTION

- When `SCI_CR1[clkz1]` is set to 0, that is, during CMOS output, the `SCI_CLK` pin does not need to connect to pull-up resistors on the board.
- When `SCI_CR1[clkz1]` is set to 1, that is, during OD output, the `SCI_CLK` pin must connect to pull-up resistors on the board.
- When `PERI_SIM_OD_CTRL` bit[2] is set to 0, that is, during CMOS output, the `SCI0_RST` pin does not need to connect to pull-up resistors on the board.
- When `PERI_SIM_OD_CTRL` bit[2] is set to 1, that is, during OD output, the `SCI0_RST` pin must connect to pull-up resistors on the board.

- When PERI_SIM_OD_CTRL bit[1] is set to 0, that is, during CMOS output, the SCI0_PWREN pin does not need to connect to pull-up resistors on the board.
- When PERI_SIM_OD_CTRL bit[1] is set to 1, that is, during OD output, the SCI0_PWREN pin must connect to pull-up resistors on the board.
- SCI_DATA must work in OD mode and connect to pull-up resistors on the board.

Figure 11-46 Typical application circuit diagram of the SCI



NOTE

Vcc1 is a 3.3 V power supply on the board, whereas Vcc is the power supply for the smart card, which is provided by the 1.8 V, 3.3 V, or 5 V power of the power management module on the board.

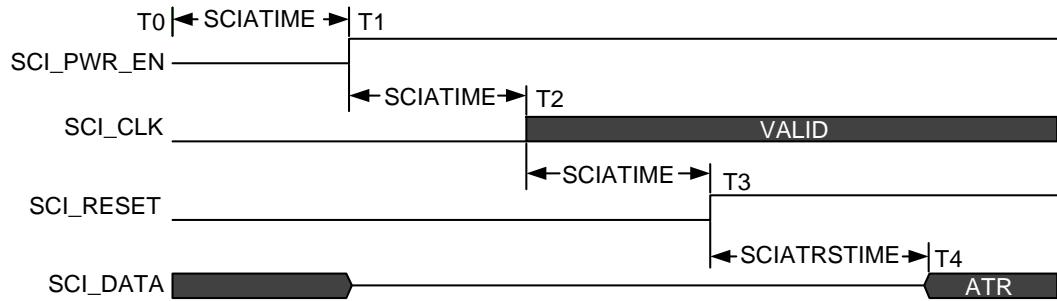
The SCI controls whether the power management module supplies power to the smart card by using the SCI_PWR_EN signal. The power management module converts the voltage between the chip and the smart card to provide the 1.8 V, 3.3 V, or 5 V voltage, meeting requirements of various smart cards. When a smart card is inserted into the slot on the board, the card detection circuit sets the SCI_DETECT signal to a valid value.

11.9.3.2 Function Implementation

Timing for Resetting the Smart Card

The SCI activates the smart card by sending the reset timing. To be specific, software writes 1 to [SCI_CR2\[startup\]](#) to complete the activation.

As shown in [Figure 11-47](#), the SCI activation timing consists of three phases. After a reset signal is sent, power is supplied at T1, the card clock is provided at T2, and the reset is deasserted at T3 to get ready to receive the answer to reset (ATR) data. After the SCI transmits the activation timing, the smart card provides ATR data at T4.

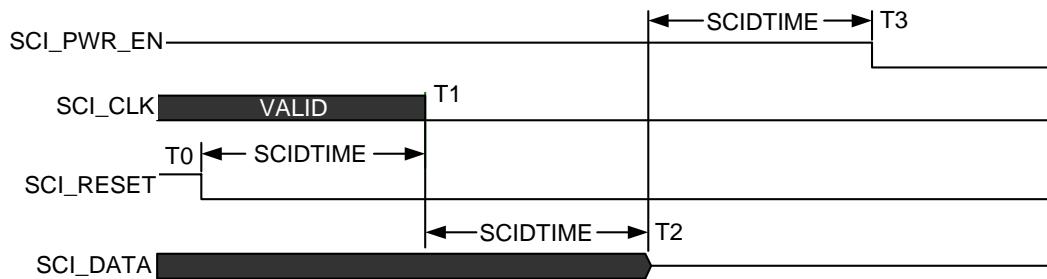
Figure 11-47 Timing for resetting the smart card **NOTE**

The delay parameter SCIATIME is defined by configuring the [SCI_ATIME](#) register.

Timing for Releasing the Smart Card

If the SCI does not receive ATR data within 400 to 40000 clock cycles of the smart card (defined in [SCI_ATRSTTIME](#)) after transmitting the activation timing, the SCI automatically starts the release timing without software intervention and transmits the ATR wait timeout interrupt ([SCI_RIS](#)[atrstoutim] is 1). After operations on the smart card are complete, the SCI releases the smart card. To be specific, software writes 1 to [SCI_CR2](#)[finish].

[Figure 11-48](#) shows the timing for releasing the smart card. The reset signal is provided at T0, the clock is cleared at T1, I/O is changed to high impedance at T2, and then the power supply is off at T3.

Figure 11-48 Timing for releasing the smart card **NOTE**

The delay parameter SCIDTIME is defined by configuring the [SCI_DTIME](#) register.

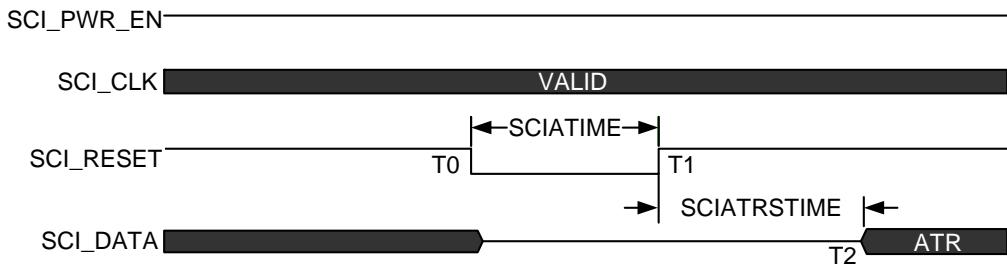
Timing for Soft-Resetting the Smart Card

If an error occurs in the received ATR data (for example, a timeout), the SCI sends a soft-reset timing to receive the ART timing again. To be specific, software writes 1 to [SCI_CR2](#)[wreset].

[Figure 11-49](#) shows the SCI timing for soft-resetting the smart card. Note that SCI_PWR_EN is always enabled.



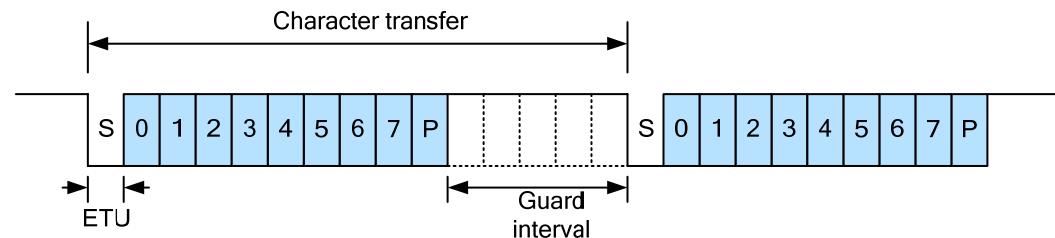
Figure 11-49 SCI timing for soft-resetting the smart card



Character Transfer Structure

In asynchronous mode, bytes are transferred between the SCI and the smart card in a unified character transfer format. A byte is transmitted as 10 consecutive ETUs. During this process, one start bit, eight data bits, and one parity bit (optional) are transmitted. When the SCI data pin is idle, high impedance is output. Figure 11-50 shows the character transfer structure supported by the SCI.

Figure 11-50 Character transfer structure supported by the SCI



S: an ETU for the duration of the start bit

P: parity bit

ETU: duration of 1-bit data



The data line in Figure 11-50 is pulled up in the idle state.

There is a guard interval after a character is transferred. During the entire guard interval, I/O (SCI_DATA) is in the high impedance state, and the SCI and smart card are in data RX state. If the RX end detects that the received data is incorrect (for example, there is a parity error), it pulls down the I/O state by one or two ETUs. After receiving the error indicator, the TX end retransmits the previous byte. This function is available only when the handshake mechanism between the SCI and the smart card is enabled.

The guard interval is configured in **SCI_CHGUARD[scichguard]** and its unit is ETU. The SCI supports the data transfer without transmitting the parity bit, which expands the compatibility of future protocols. The function can be implemented by setting **SCI_CR0[paritybit]** to 0.



11.9.4 Operating Mode

11.9.4.1 Interaction Between the SCI and a Smart Card

The interaction between the SCI and a smart card is as follows:

- Step 1** Initialize the SCI module. The smart card clock, initial operating mode of the SCI, and interrupt enable bits need to be configured by configuring registers including [SCI_CR0](#), [SCI_CR1](#), [SCI_CLKICC](#), [SCI_VALUE](#), [SCI_BAUD](#), [SCI_STABLE](#), threshold time registers, and [SCI_IMSC](#).
- Step 2** Insert a smart card. Then [SCI_DETECT](#) is valid. The SCI reports an interrupt after detecting card insertion.
- Step 3** Software writes 1 to [SCI_CR2](#)[startup]. The SCI resets and activates the smart card. See [Figure 11-47](#).
- Step 4** The smart card sends an ATR to start information exchange between the smart card and the SCI.
- Step 5** Software reads data from the RX FIFO and reconfigures the SCI parameters based on the read ATR information. The parameters include the ETU, smart card clock rate, and forward convention or inverse convention.
- Step 6** The SCI and the smart card exchange data in asynchronous half-duplex mode. Data can be written to the TX FIFO only when software switches to the TX mode. Data can be read to the RX FIFO only when software switches to the RX mode. For details about the character structure, see "[Character Transfer Structure](#)."
- Step 7** The SCI sends a release timing (see [Figure 11-48](#)) to release the smart card after data transfer is complete.

----End

11.9.4.2 Configurations of the Smart Card Clock and ETU



CAUTION

- During ATR reception, the frequency of the smart card clock ranges from 1 MHz to - 5MHz and the ETU is $372/F_{SCL_CLK}$ according to the T0 and T1 protocols. According to the T14 protocol, the smart card clock is 6 MHz at the maximum and the ETU is $620/F_{SCL_CLK}$.
- After the ATR is received, the SCI needs to configure the ETU based on the F and D dividers in the ART: $ETU = F/F_{SCL_CLK} \times 1/D$.
- [SCI_VALUE](#) and [SCI_BAUD](#) must be configured based on the following formula:
$$(1 + Baud) \times Value/F_{REFCLK} = F/F_{SCL_CLK} \times 1/D$$

For details about how to configure the smart card clock, see section [11.9.4.1 "Interaction Between the SCI and a Smart Card"](#).

The ETU is configured in two phases:

- Phase 1: Step 1 in section [11.9.4.1 "Interaction Between the SCI and a Smart Card"](#), including the ETU configuration before the ATR timing is received.



- Phase 2: Step 5 in section [11.9.4.1 "Interaction Between the SCI and a Smart Card,"](#) including the ETU reconfiguration after the parameters D and F in the ATR timing are received.

The frequency of SCI_CLK and level width of [SCI_DATA](#) can be configured by configuring registers. For details about the formula for calculating the values, see the descriptions of [SCI_VALUE](#) and [SCI_CLKICC](#).

NOTE

In the formulas described in this section, FSCI_CLK is the frequency of SCI_CLK and its unit is MHz.

The ETU is calculated as follows: $ETU = (1 + \text{Baud}) \times \text{Value}/F_{REFCLK}$. For details, see the description of [SCI_VALUE](#).

11.9.5 Register Summary

Hi3796M V100 has one SCI unit.

[Table 11-18](#) describes SCI registers.

Table 11-18 Summary of SCI registers (base address: 0xF8B1_8000)

Offset Address	Register	Description	Page
0x0000	SCI_DATA	SCI data register	
0x0004	SCI_CR0	SCI control register 0	
0x0008	SCI_CR1	SCI control register 1	
0x000C	SCI_CR2	SCI control register 2	
0x0010	SCI_CLKICC	Smart card clock frequency register	
0x0014	SCI_VALUE	SCI_BAUD cycle count register in an ETU	
0x0018	SCI_BAUD	Baud rate clock frequency divider register	
0x001C	SCI_TIDE	TX or RX FIFO overflow threshold register	
0x0024	SCI_STABLE	Smart card insertion stable time register	
0x0028	SCI_ATIME	Smart card activation time register	
0x002C	SCI_DTIME	Smart card release time register	
0x0030	SCI_ATRSTIME	ATR RX wait time threshold register	
0x0034	SCI_ATRDTIME	ATR RX time threshold register	
0x0038	SCI_STOPTIME	Clock stop time register	
0x003C	SCI_STARTTIME	Smart card clock recovery time register	
0x0040	SCI_RETRY	TX or RX retry times register	
0x0044	SCI_CHTIMELS	Lower 16-bit character RX interval timeout threshold register	



Offset Address	Register	Description	Page
0x0048	SCI_CHTIMEMS	Upper 16-bit character RX interval timeout threshold register	
0x004C	SCI_BLKTIMELS	Lower 16-bit block RX interval timeout threshold register	
0x0050	SCI_BLKTIMEMS	Upper 16-bit block RX interval timeout threshold register	
0x0054	SCI_CHGUARD	Character guard interval register	
0x0058	SCI_BLKGUARD	Block guard interval register	
0x005C	SCI_RXTIME	RX FIFO read timeout threshold register	
0x0060	SCI_FIFOSTATUS	FIFO status register	
0x0064	SCI_TXCOUNT	TX FIFO data count register	
0x0068	SCI_RXCOUNT	RX FIFO data count register	
0x006C	SCI_IMSC	Interrupt mask register	
0x0070	SCI_RIS	Raw interrupt register	
0x0074	SCI_MIS	Masked interrupt register	
0x0078	SCI_ICR	Interrupt clear register	
0x007C	SCI_SYNCACT	Activation register in sync mode	
0x0080	SCI_SYNCTX	TX clock and data stream register in sync mode	
0x0084	SCI_SYNCRX	RX clock and data stream register in sync mode	

11.9.6 Register Description

SCI_DATA

SCI_DATA is an SCI data register. It is used to transmit or receive characters and serves as the data interface between the SCI and software.

The software can write to the register only after [SCI_CR1\[mode\]](#) is set to 1. If [SCI_CR1\[mode\]](#) is set to 0, writing to this register has no effect.



Offset Address										Register Name						Total Reset Value					
Bit	0x0000					SCI_DATA					0x0000										
Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
reserved									parity	scidata											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
Bits	Access	Name			Description																
[15:9]	-	reserved			Reserved																
[8]	RO	parity			Parity check error flag This bit is transmitted with data during data transmission. During data reception, software checks whether parity check is correct based on the value of this bit. 0: correct 1: incorrect																
[7:0]	RW	scidata			8-bit data to be read or written																

SCI_CR0

SCI_CR0 is SCI control register 0. It is used to control the information such as clock enable, parity check, transfer handshake, and transfer bit definition.



CAUTION

- SCI_CR0[paritybit] controls whether to enable the parity check function during data transmission or reception. When this bit is set to 0, the received or transmitted frame does not contain the parity bit. According to the T0 or T1 protocol, this bit must be set to 1.
- SCI_CR0[clkdis] and SCI_CR0[clkval] control whether to enable the smart card clock and configure the state of the clock pin after the clock is disabled.
- SCI_CR0[rxnak] and SCI_CR0[txnak] control whether to enable the handshake mechanisms between the SCI and the smart card. A handshake is implemented when the RX end pulls down the data line (I/O) to request the TX end to retransmit characters after detecting a parity check error in the data sent from the TX end. The handshake mechanisms for data transmission and reception are separately enabled. The maximum retry count during data transmission or reception is defined by [SCI_RETRY](#). During ATR reception, character retransmission is not allowed; therefore, the handshake mechanism must be disabled by setting the corresponding TX or RX control bit to 0.
- SCI_CR0[rxparity] and SCI_CR0[txparity] control the parity check mode during data reception and transmission respectively.



According to the direct convention, the low level of the data line (I/O) indicates logic 0 and the LSB of data is after the start bit of a frame. If SCI_CR0 bit[1:0] is set to 0b00, the direct convention is selected.

According to the inverse convention, the low level of the data line (I/O) indicates logic 1 and the MSB of data is after the start bit of a frame. If SCI_CR0 bit[1:0] is set to 0b11, the inverse convention is selected.

NOTE

The character bit sequence and inversion of the data and parity bits can be separately configured. These features enable the SCI to support non-standard protocols (non-direct convention or non-inverse convention).

Before the start TS character of the ATR is received, SCI_CR0 bit[1:0] must be set to 0b00.

	Offset Address					Register Name					Total Reset Value					
	0x0004					SCI_CR0					0x0000					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					detect_inv	vccen_inv	paritybit	clkval	clkdis	rxnak	rxparity	txnak	txparity	order	sense
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description												
[15:11]	-	reserved		Reserved												
[10]	RW	detect_inv		Valid level for the detected input signal 0: active high 1: active low												
[9]	RW	vccen_inv		Valid level for the vccen output signal 0: active high 1: active low												
[8]	RW	paritybit		Parity bit transfer enable 0: disabled 1: enabled												
[7]	RW	clkval		Status of the smart card clock when the clock stops 0: The smart card clock retains low level. 1: The smart card clock retains high level.												
[6]	RW	clkdis		Clock start/stop control 0: The clock starts. 1: The clock stops.												



[5]	RW	rxnak	Behavior control in RX mode 0: The SCI does not pull down the I/O line when detecting a parity error. 1: The SCI pulls down the I/O line when detecting a parity error.
[4]	RW	rxparity	Parity check mode control in RX mode 0: even parity check 1: odd parity check
[3]	RW	txnak	Behavior control in TX mode 0: The SCI does not detect whether the receiver pulls down the I/O line. 1: The SCI detects whether the receiver pulls down the I/O line.
[2]	RW	txparity	Parity check mode control in TX mode 0: even parity check 1: odd parity check
[1]	RW	order	Character bit sequence 0: The LSB is transmitted or received after the start bit. 1: The MSB is transmitted or received after the start bit.
[0]	RW	sense	Whether to detect the data and parity bits on the I/O line after inversion 0: The SCI directly detects data and parity bits on the I/O line. 1: The SCI detects data and parity bits on the I/O line after inversion.

SCI_CR1

SCI_CR1 is SCI control register 1. It is used to control the transfer mode, TX/RX mode, block timeout, and ATR timeout of synchronous and asynchronous cards.

Offset Address		Register Name													Total Reset Value				
0x0008		SCI_CR1													0x0000				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Name	reserved												syncard	exdbnce	bgtcn	clkz1	mode	blkcn	atrdn
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bits	Access	Name		Description															
[15:7]	-	reserved		Reserved															



[6]	RW	synccard	Operation mode of the smart card 0: async mode 1: sync mode
[5]	RW	exdbnce	Debounce wait counter select after the card is inserted and stable. For details, see the description of SCI_STABLE. 0: The entire internal debounce counter is used. 1: The non-programmable part of the internal counter is bypassed.
[4]	RW	bgtcn	Block guard counter enable for the block guard mechanism 0: disabled 1: enabled
[3]	RW	clkz1	SCI_CLK pin output configuration 0: CMOS output 1: OD output
[2]	RW	mode	TX/RX mode 0: RX mode 1: TX mode
[1]	RW	blkcn	Block timeout count mechanism enable 0: disabled 1: enabled
[0]	RW	atrden	ATR timeout count mechanism enable 0: disabled 1: enabled

SCI_CR2

SCI_CR2 is SCI control register 2. After a value is written to a specific bit of SCI_CR2, the SCI activates, releases, or soft-resets the smart card. When the SCI is releasing the smart card, write operations on this register are ignored. In other cases, writing 1 to SCI_CR2[finish] enables the SCI to immediately send a card release timing.



CAUTION

- The software writes to SCI_CR2 only in appropriate card operation phases. An inappropriate write may result in an unexpected result.
- The smart card can be soft-reset by software only after the card is activated.



	Offset Address										Register Name				Total Reset Value			
	0x000C										SCI_CR2				0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	reserved												wreset	finish	startup			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bits	Access	Name		Description														
[15:3]	-	reserved		Reserved														
[2]	WO	wreset		Soft reset on the smart card 0: invalid 1: soft-reset Note: This bit can be written only after the activation timing is transmitted. Otherwise, writing to this bit has no effect.														
[1]	WO	finish		Release operation on the smart card 0: invalid 1: released Note: This bit can be written only after the card reader detects a smart card. Otherwise, writing to this bit has no effect.														
[0]	WO	startup		Activation operation on the smart card 0: invalid 1: activated Note: This bit can be written only after the card reader detects a smart card. Otherwise, writing to this bit has no effect.														

SCI_CLKICC

SCI_CLKICC is a smart card clock frequency register. It defines the clock divider of the external smart card. The frequency of the smart card clock is calculated as follows: $F_{SCI_CLK} = F_{REFCLK}/2 \times (clkicc + 1)$. F_{SCI_CLK} indicates the clock frequency, F_{REFCLK} indicates the frequency of the reference clock, and $clkicc$ is the value configured in SCI_CLKICC[clkicc].

	Offset Address										Register Name				Total Reset Value			
	0x0010										SCI_CLKICC				0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Name	reserved										clkicc							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bits	Access	Name		Description														
[15:8]	-	reserved		Reserved														



[7:0]	RW	clkicc	Divider of the smart card clock (obtained by dividing the reference clock frequency). Its value range is 0–255.
-------	----	--------	---

SCI_VALUE

SCI_VALUE is an SCI_BAUD cycle count register in an ETU. Its value range is 5–255 and is used to calculate the ETU as follows: ETU = (1 + baud) x value/FREFCLK. FREFCLK is the frequency of the reference clock, baud is the value configured in SCI_BAUD[baud], and value is the value configured in SCI_VALUE[value].

	Offset Address								Register Name								Total Reset Value							
	0x0014								SCI_VALUE								0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	reserved															value								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
Bits	Access	Name			Description																			
[15:8]	-	reserved			Reserved																			
[7:0]	RW	value			Number of SCI_BAUD cycles in an ETU																			

SCI_BAUD

SCI_BAUD is a baud rate clock divider register. It is used to calculate the ETU and its value range is 0x1–0xFFFF. For details about how to calculate the ETU, see the description of [SCI_BAUD](#).

	Offset Address								Register Name								Total Reset Value							
	0x0018								SCI_BAUD								0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	baud															baud								
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			Clock divider of the baud rate																			

SCI_TIDE

SCI_TIDE is a TX/RX FIFO overflow threshold register.



NOTE

The characters in the TX FIFO are removed only after data is successfully transmitted.



	Offset Address					Register Name					Total Reset Value					
	0x001C					SCI_TIDE					0x0000					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				rxtide					txtide						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name			Description											
[15:12]	-	reserved			Reserved											
[11:6]	RW	rxtide			Value for triggering the SCIRXTIDEINTR interrupt When the number of characters in the RX FIFO is greater than or equal to SCI_TIDE[rxtide], the RX FIFO overflow interrupt is triggered.											
[5:0]	RW	txtide			Value for triggering the SCI TXTIDEINTR interrupt When the number of characters in the TX FIFO is less than or equal to SCI_TIDE[txtide], the TX FIFO overflow interrupt is triggered.											

SCI_STABLE

SCI_STABLE is a smart card insertion stable time register. After the SCI detects that the SCI_DETECT signal is valid, the signal must retain valid for the period (in second) specified by T_{STABLE} that corresponds to the value defined in the SCI_STABLE register. After the period specified by T_{STABLE} has elapsed, an interrupt is triggered, indicating that the smart card has been properly inserted. The value of T_{STABLE} varies according to counter:

- When the entire internal debounce counter is selected (that is, SCI_CR1[exdbnse] is set to 0), T_{STABLE} is calculated as follows: T_{STABLE} = (1 + Stable) x 65535 x T_{REFCLK}. T_{REFCLK} indicates the SCI reference clock cycle (1/48 MHz), and stable is the value configured in SCI_STABLE[stable].
- When the non-programmable part of the internal debounce counter is bypassed (that is, SCI_CR1[exdbnse] is set to 1), T_{STABLE} is calculated as follows: T_{STABLE} = Stable x T_{REFCLK}.

	Offset Address					Register Name					Total Reset Value					
	0x0024					SCI_STABLE					0x0000					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved					stable										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name			Description											
[15:10]	-	reserved			Reserved											
[9:0]	RW	stable			Duration during which the card detection signal is held high											



SCI_ATIME

SCI_ATIME is a smart card activation time register. SCI_ATIME[etime] corresponds to SCIAETIME in [Figure 11-47](#) and indicates the duration of each of the three phases involved in card reset and activation.

NOTE

SCI_ATIME[etime] must be at least 40,000 smart card clock (SCI_CLK) cycles to ensure card reset. SCI_ATIME[etime] must also ensure power stability for the smart card.

Offset Address										Register Name						Total Reset Value					
0x0028										SCI_ATIME						0x0000					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name																					
Reset																					
Bits	Access		Name			Description															
[15:0]	RW		etime			Duration (SCI_CLK cycle count) of each of the three phases involved in the card activation timing. For details about the timing for resetting the smart card, see Figure 11-47 .															

SCI_DTIME

SCI_DTIME is a smart card release time register. SCI_DTIME[dtime] corresponds to SCIDTIME in [Figure 11-48](#). It indicates the duration of each of the three phases involved in card release.

Offset Address										Register Name						Total Reset Value					
0x002C										SCI_DTIME						0x0000					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Name																					
Reset																					
Bits	Access		Name			Description															
[15:0]	RW		dtime			Duration (SCI reference clock cycle) of each of the three phases involved in the card release timing. For details about the timing for releasing the smart card, see Figure 11-48 .															

SCI_ATRSTIME

SCI_ATRSTIME is an ATR RX start time threshold register. SCI_ATRSTIME[dtime] corresponds to SCIAINTRSTIME in [Figure 11-47](#). Based on the smart card clock, the register defines the time threshold from finishing smart card reset to starting to receive the first ATR character. If waiting for the ATR RX start signal times out, an interrupt is triggered and [SCI_RIS](#)[atrstoutim] is set to 1.



SCI_ATRSTIME[atrstime] must be set to 40,000 SCI_CLK cycles by software.

Offset Address										Register Name					Total Reset Value		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	atrstime																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bits	Access	Name	Description														
[15:0]	RW	atrstime	Timeout threshold (SCI_CLK cycle) for starting ATR reception														

SCI_ATRDTIME

SCI_ATRDTIME is an ATR RX time threshold register. If the duration from receiving the first ATR character to receiving the last one exceeds the duration (in ETU) defined in this register, an interrupt is triggered and SCI_RIS[atrdtoutris] is set to 1.

NOTE

- Before using this register, set SCI_CR1[atrden] to 1 to enable the timeout count mechanism.
- According to the protocol, it is recommended that this register be set to 19,200 ETUs (0x4B00).

Offset Address										Register Name					Total Reset Value		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	atrdtime																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bits	Access	Name	Description														
[15:0]	RW	atrdtime	ATR RX timeout threshold (in ETU), counting from the start bit of the first ATR character														

SCI_STOPTIME

SCI_STOPTIME is a clock stop time register. When the smart card clock stops, it becomes invalid after the time defined in SCI_STOPTIME (SCI_CLK cycles) has elapsed. Then an interrupt is triggered, and SCI_RIS[clkstpris] is set to 1.

According to the protocol, the minimum clock stop time is 1860 SCI_CLK cycles, that is, when SCI_STOPTIME[stoptime] is set to 0x744.



Offset Address												Register Name	Total Reset Value			
0x0038												SCI_STOPTIME	0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				stoptime											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name			Description											
[15:12]	-	reserved			Reserved											
[11:0]	RW	stoptime			Time (SCI_CLK cycles) for stopping the smart card clock											

SCI_STARTTIME

SCI_STARTTIME is a smart card clock recovery time register. If clock recovery is enabled, data can be transferred between the smart card and the SCI after the time (SCI_CLK cycles) defined in SCI_STARTTIME has elapsed. Then, an interrupt is triggered, and [SCI_RIS\[clkactris\]](#) is set to 1.

According to the protocol, the minimum clock recovery time is 700 SCI_CLK cycles, that is, when SCI_STARTTIME[starttimr] is set to 0x2BC.

Offset Address												Register Name	Total Reset Value			
0x003C												SCI_STARTTIME	0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved				starttime											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name			Description											
[15:12]	-	reserved			Reserved											
[11:0]	RW	starttime			Duration (SCI_CLK cycles) from starting the card clock to activating data transfer between the SCI and the smart card											

SCI_RETRY

SCI_RETRY is an RX/TX retry times register. It defines the allowed number of TX or RX retry times. The details are as follows:

- If SCI_CR0[txnak] is enabled, the SCI checks whether the RX end identifies a parity error. SCI_RETRY[rxretry] defines the maximum number of retry times after a parity error occurs. If the maximum number of retry times is exceeded, an interrupt is triggered and [SCI_RIS\[txerrris\]](#) is set to 1.
- If SCI_CR0[rxnak] is enabled, the SCI checks whether a parity error occurs in the received data. SCI_RETRY[txretry] defines the maximum number of retry times allowed after a parity error occurs. If the number is exceeded but the RX error still persists, [SCI_DATA\[parity\]](#) is set to 1.



	Offset Address								Register Name				Total Reset Value			
	0x0040								SCI_RETRY				0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved								rxretry				txretry			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description												
[15:6]	-	reserved		Reserved												
[5:3]	RW	rxretry		Maximum number of RX retry times after a parity error occurs during reception												
[2:0]	RW	txretry		Maximum number of TX retry times after a parity error occurs during transmission												

SCI_CHTIMELS

SCI_CHTIMELS is a character RX interval timeout threshold lower 16-bit register.



CAUTION

The SCI_CHTIME register consists of the lower 16-bit SCI_CHTIMELS register and the upper 16-bit SCI_CHTIMEMS register. SCI_CHTIME does not immediately take effect after software writes to SCI_CHTIMEMS. It takes effect only after software writes to SCI_CHTIMELS. Therefore, you need to configure the upper 16 bits and then the lower 16 bits of SCI_CHTIME in sequence.

SCI_CHTIMELS is a character RX interval timeout threshold lower 16-bit register.

SCI_CHTIME defines the maximum time interval (in ETU) between the start edges of two consecutive characters received from the smart card. If the character RX interval times out, an interrupt is triggered and SCI_RIS[chtoutris] is set to 1.

	Offset Address								Register Name				Total Reset Value			
	0x0044								SCI_CHTIMELS				0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	chtimels															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description												
[15:0]	RW	chtimels		Lower 16 bits of the character RX interval timeout threshold register SCICHTIME												



SCI_CHTIMEMS

SCI_CHTIMEMS is a character RX interval timeout threshold upper 16-bit register.

	Offset Address								Register Name								Total Reset Value														
	0x0048								SCI_CHTIMEMS								0x0000														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
Name	chtimems																														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
Bits	Access	Name		Description																											
[15:0]	RW	chtimems		Upper 16 bits of the character RX interval timeout threshold register SCICHTIME																											

SCI_BLKTIMELS

SCI_BLKTIMELS is a block RX interval timeout threshold lower 16-bit register.



CAUTION

The SCI_BLKTIME register consists of the lower 16-bit SCI_BLKTIMELS register and the upper 16-bit SCI_BLKTIMEMS register. SCI_BLKTIME does not immediately take effect after software writes to SCI_BLKTIMEMS. It takes effect only after software writes to SCI_BLKTIMELS. Therefore, you need to configure the upper 16 bits and then the lower 16 bits of SCI_BLKTIME in sequence.

SCI_BLKTIMELS is a block RX interval timeout threshold lower 16-bit register.

SCI_BLKTIME defines the maximum block RX interval (in ETU). The interval is counted from the start edge of the last character that is transmitted to the smart card to the time when the first character returned from the smart card is received.

If the interval times out, an interrupt is triggered and SCI_RIS[blkoutim] is set to 1.

	Offset Address								Register Name								Total Reset Value														
	0x004C								SCI_BLKTIMELS								0x0000														
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
Name	blktimels																														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
Bits	Access	Name		Description																											
[15:0]	RW	blktimels		Lower 16 bits of the block RX interval timeout threshold register SCIBLKTIME																											



SCI_BLKTIMEMS

SCI_BLKTIMES is a block RX interval timeout threshold upper 16-bit register.

SCI_CHGUARD

SCI_CHGUARD is a character guard interval register. It defines the minimum extra guard interval (in ETU) between the start edges of two consecutive characters when the SCI is transmitting characters to the smart card. The character guard interval depends on the global interface byte TC1 (obtained from the ATR timing) defined in the protocol.

Offset Address		Register Name		Total Reset Value		
0x0054		SCI_CHGUARD		0x0000		
Bit	15 14	13 12	11 10	9 8	7 6 5 4 3 2 1 0	
Name	reserved		scichguard			
Reset	0 0	0 0	0 0	0 0	0 0 0 0 0 0 0 0	
Bits	Access	Name	Description			
[15:8]	-	reserved	Reserved			
[7:0]	RW	scichguard	Character guard interval (in ETU)			

SCI BLKGUARD

SCI_BLKGUARD is a block guard interval register. When the transfer directions of two consecutive characters are opposite, the interval between their start edges is the value defined in SCI_BLKGUARD.



	Offset Address								Register Name								Total Reset Value							
	0x0058								SCI_BLKGUARD								0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	reserved												sciblkguard											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
Bits	Access	Name			Description																			
[15:8]	-	reserved			Reserved																			
[7:0]	RW	sciblkguard			Minimum interval (unit: ETU) between two consecutive characters with opposite transfer directions																			

SCI_RXTIME

SCI_RXTIME is an RX FIFO read timeout threshold register. When there are characters in the RX FIFO and the characters are not read within the period defined in SCI_RXTIME, the RX read timeout interrupt is triggered and [SCI_RIS](#)[rtoutris] is set to 1.



CAUTION

Never set SCI_RXTIME[rxtime] to 0x0000 when you use the read timeout interrupt function of the RX FIFO.

	Offset Address								Register Name								Total Reset Value							
	0x005C								SCI_RXTIME								0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	rxtime																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
Bits	Access	Name			Description																			
[15:0]	RW	rxtime			RX read timeout threshold (unit: SCI_CLK cycle)																			

SCI_FIFOSTATUS

SCI_FIFOSTATUS is a FIFO status register.



	Offset Address								Register Name								Total Reset Value			
	0x0060								SCI_FIFOSTATUS								0x000A			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Name	reserved												txfe	txff	txfe	txff				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0				
Bits	Access	Name		Description																
[15:4]	-	reserved		Reserved																
[3]	RO	rxfe		RX FIFO empty status 0: not empty 1: empty																
[2]	RO	txff		RX FIFO full status 0: not full 1: full																
[1]	RO	txfe		TX FIFO empty status 0: not empty 1: empty																
[0]	RO	txff		TX FIFO full status 0: not full 1: full																

SCI_TXCOUNT

SCI_TXCOUNT is a TX FIFO data count register.



CAUTION

If an error defined in the T0 protocol occurs when characters are transmitted to the smart card, and the number of retransmission times exceeds the allowed value defined in SCI_RETRY, an interrupt is triggered and [SCI_RIS\[txerrris\]](#) is set to 1. Before the next transmission, the TX FIFO must be cleared by writing to SCI_TXCOUNT.



SCI_RXCOUNT

SCI_RXCOUNT is an RX FIFO data count register.

SCI IMSC

SCI IMSC is an interrupt mask register.



		Offset Address 0x006C						Register Name SCI_IMSC						Total Reset Value 0x0000				
Bit	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	reserved		txtideim	rxtideim	clkactim	clkstpim	rorim	rtoutim	chtoutim	blktoutim	atrdtoutim	atrstoutim	txerrim	carddnum	cardupim	cardoutim	cardinim	
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name			Description													
[15]	-	reserved			Reserved													
[14]	RW	txtideim			TX FIFO overflow interrupt mask 0: masked 1: not masked													
[13]	RW	rxtideim			RX FIFO overflow interrupt mask 0: masked 1: not masked													
[12]	RW	clkactim			Smart card clock recovery interrupt mask 0: masked 1: not masked													
[11]	RW	clkstpim			Smart card clock stop interrupt mask 0: masked 1: not masked													
[10]	RW	rorim			RX overload interrupt mask 0: masked 1: not masked													
[9]	RW	rtoutim			Read timeout interrupt mask 0: masked 1: not masked													
[8]	RW	chtoutim			Character interval timeout interrupt mask 0: masked 1: not masked													
[7]	RW	blktoutim			Block interval timeout interrupt mask 0: masked 1: not masked													
[6]	RW	atrdtoutim			ATR RX timeout interrupt mask 0: masked 1: not masked													



[5]	RW	atrstoutim	ATR wait timeout interrupt mask 0: masked 1: not masked
[4]	RW	txerrim	TX error interrupt mask 0: masked 1: not masked
[3]	RW	carddnim	Smart card release interrupt mask 0: masked 1: not masked
[2]	RW	cardupim	Smart card activate interrupt mask 0: masked 1: not masked
[1]	RW	cardoutim	Smart card remove interrupt mask 0: masked 1: not masked
[0]	RW	ardinim	Smart card insert interrupt mask 0: masked 1: not masked

SCI_RIS

SCI_RIS is a raw interrupt register. This register defines the raw interrupts that are not masked.

		Offset Address		Register Name										Total Reset Value		
		0x0070		SCI_RIS										0x400A		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved	txtideris	rxtideris	clkactris	clkspiris	rroris	ritouris	chtoutris	blktoutris	atrdtoutris	atrstoutris	txerris	carddnris	cardupris	cardoutris	ardinris
Reset	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0
Bits	Access	Name			Description											
[15]	-	reserved			Reserved											



[14]	RO	txtideris	Raw TX FIFO overflow interrupt. This bit is set to 1 if the number of characters in the TX FIFO is less than or equal to the threshold. 0: No interrupt is generated. 1: An interrupt is generated.
[13]	RO	rxtideris	Raw RX FIFO overflow interrupt. This bit is set to 1 if the number of characters in the RX FIFO is greater than or equal to the threshold. 0: No interrupt is generated. 1: An interrupt is generated.
[12]	RO	clkactris	Raw smart card clock recovery interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[11]	RO	clkstpris	Raw smart card clock stop interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[10]	RO	rrorris	Raw RX overload interrupt. This bit is set to 1 if the RX FIFO is full and new characters are received. 0: No interrupt is generated. 1: An interrupt is generated.
[9]	RO	rtoutris	Raw read timeout interrupt. This bit is set to 1 if the CPU does not fetch data in the RX FIFO within the specified period. 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	chtoutris	Raw character interval timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	blktooutris	Raw block interval timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[6]	RO	atrdtoutris	Raw ATR RX timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	atrstoutris	Raw ATR wait timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	txerrris	Raw TX error interrupt. This bit is set to 1 if an error occurs in the transmitted characters and the error still persists after a specified number of retries. 0: No interrupt is generated. 1: An interrupt is generated.



[3]	RO	carddnris	Raw smart card release interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	cardupris	Raw smart card activate interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	cardoutris	Raw smart card remove interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	ardinris	Raw smart card insert interrupt 0: No interrupt is generated. 1: An interrupt is generated.

SCI_MIS

SCI_MIS is a masked interrupt register. This register defines the results obtained after raw interrupts are masked.

	Offset Address 0x0074								Register Name SCI_MIS								Total Reset Value 0x0000							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Name	reserved	txtidemis	rxtidemis	clkactmis	clkstpmis	rormis	rtoutmis	chtoutmis	blkoutmis	atrdoutmis	atrstoutim	txermis	carddnmis	cardupmis	cardoutmis	cardnmis								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
Bits	Access	Name		Description																				
[15]	-	reserved		Reserved																				
[14]	RO	txtidemis		Masked TX FIFO overflow interrupt 0: No interrupt is generated. 1: An interrupt is generated.																				
[13]	RO	rxtidemis		Masked RX FIFO overflow interrupt 0: No interrupt is generated. 1: An interrupt is generated.																				
[12]	RO	clkactmis		Masked smart card clock recovery interrupt 0: No interrupt is generated. 1: An interrupt is generated.																				



[11]	RO	clkstpmis	Masked smart card clock stop interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[10]	RO	rormis	Masked RX overload interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[9]	RO	rtoutmis	Masked read timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[8]	RO	chtoutmis	Masked character interval timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[7]	RO	blktoutmis	Masked block interval timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[6]	RO	atrdtoutmis	Masked ATR RX timeout interrupt status 0: No interrupt is generated. 1: An interrupt is generated.
[5]	RO	atrstoutim	Masked ATR wait timeout interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[4]	RO	txerrmis	Masked TX error interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[3]	RO	carddnmis	Masked smart card release interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[2]	RO	cardupmis	Masked smart card activate interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[1]	RO	cardoutmis	Masked smart card remove interrupt 0: No interrupt is generated. 1: An interrupt is generated.
[0]	RO	ardinmis	Masked smart card insert interrupt 0: No interrupt is generated. 1: An interrupt is generated.



SCI_ICR

SCI_ICR is an interrupt clear register.

	Offset Address				Register Name								Total Reset Value			
	0x0078				SCI_ICR								0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	reserved		clkactic	clkstpic	roric	rtoutic	chtoutic	blktoutic	atrdtouic	atrstoutic	txerric	cardnic	cardupic	cardoutic	cardinic	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name		Description												
[15:13]	-	reserved		Reserved												
[12]	WO	clkactic		Smart card clock recovery interrupt clear 0: not cleared 1: cleared												
[11]	WO	clkstpic		Smart card clock stop interrupt clear 0: not cleared 1: cleared												
[10]	WO	roric		RX overload interrupt clear 0: not cleared 1: cleared												
[9]	WO	rtoutic		Read timeout interrupt clear 0: not cleared 1: cleared												
[8]	WO	chtoutic		Character interval timeout interrupt clear 0: not cleared 1: cleared												
[7]	WO	blktoutic		Block interval timeout interrupt clear 0: not cleared 1: cleared												
[6]	WO	atrdtouic		ATR RX timeout interrupt clear 0: not cleared 1: cleared												
[5]	WO	atrstoutic		ATR wait timeout interrupt clear 0: not cleared 1: cleared												



[4]	WO	txerric	TX error interrupt clear 0: not cleared 1: cleared
[3]	WO	carddnic	Smart card release interrupt clear 0: not cleared 1: cleared
[2]	WO	cardupic	Smart card activate interrupt clear 0: not cleared 1: cleared
[1]	WO	cardoutic	Smart card remove interrupt clear 0: not cleared 1: cleared
[0]	WO	cardinic	Smart card insert interrupt clear 0: not cleared 1: cleared

SCI_SYNCACT

SCI_SYNCACT is an activation register in sync mode. The register is used to enable data and clocks, implement reset, and control power supply in sync mode. The register also provides status information. The corresponding status bits are automatically updated during activation, release, or warm reset.

Offset Address 0x007C										Register Name SCI_SYNCACT							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved						cardpresent	nscidataen	nscidataouten	scilckout	nsciclkken	rsciclkouten	fcb	dataen	clken	reset	power
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	Access	Name			Description												
[15:11]	-	reserved			Reserved												
[10]	RO	cardpresent			Smart card presence 0: No smart card is detected. 1: A smart card is detected.												



[9]	RO	nscidataen	Tristate control enable for the off-chip buffer of data 0: enabled 1: disabled
[8]	RO	nscidataouten	Tristate control enable for the data buffer 0: enabled 1: disabled
[7]	RO	sciclkout	SCI clock output enable 0: enabled 1: disabled
[6]	RO	nsciclknen	Tristate control enable for the off-chip buffer of the clock 0: enabled 1: disabled
[5]	RO	nsciclkouten	Tristate control enable for the data buffer 0: enabled 1: disabled
[4]	RW	fcb	Functional code. This bit works with the creset bit to indicate the type of a command to be executed.
[3]	RW	dataen	SCI data output enable 0: disabled 1: enabled
[2]	RW	clken	SCI clock output enable 0: disabled 1: enabled
[1]	RW	creset	Smart card reset signal control enable 0: enabled 1: disabled
[0]	RW	power	Card power (VCC) enable 0: Enabled. 1: disabled

SCI_SYNCTX

SCI_SYNCTX is a TX clock and data stream register in sync mode. It is used to determine whether to transmit clocks and data in sync mode.



Offset Address 0x0080										Register Name SCI_SYNCTX				Total Reset Value 0x0000			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Name	reserved										wfcb	wrst	wclken	wdataen	wclk	wdata	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name		Description													
[15:6]	-	reserved		Reserved													
[5]	RW	wfcb		Functional code in sync mode. The bit works with the wrst bit to indicate the type of a command to be executed.													
[4]	RW	wrst		Card reset signal enable in sync mode 0: disabled 1: enabled													
[3]	RW	wclken		Tristate control enable for the off-chip buffer of the clock in sync mode 0: enabled 1: disabled													
[2]	RW	wdataen		Card data output enable in sync mode 0: disabled 1: enabled													
[1]	RW	wclk		SCI clock enable in sync mode 0: enabled 1: disabled													
[0]	RW	wdata		SCI data enable in sync mode 0: enabled 1: disabled													

SCI_SYNCRX

SCI_SYNCRX is an RX clock and data stream register in sync mode. It is used to determine whether to receive clocks and data in sync mode.



Offset Address										Register Name					Total Reset Value					
Bit	0x0084					SCI_SYNCRX					0x0000									
Name	reserved														rclk	rdata				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bits	Access	Name		Description																
[15:2]	-	reserved		Reserved																
[1]	RO	rclk		Raw clock																
[0]	RO	rdata		Raw data																



Contents

A Acronyms and Abbreviations.....	A-1
-----------------------------------	-----



A Acronyms and Abbreviations

A

AAC	advanced audio coding
AAF	anti-aliasing filter
ABR	average bit rate
AC	alternating current
ACA	accessory charge adapter
ACC	automatic contrast control
ACD	auto command done
ACM	adaptive coding and modulation
ADP	attach detection protocol
ADC	analog-to-digital converter
AE	automatic exposure
AEC	audio echo cancellation
AES	advanced encryption standard
AF	adaption field
AGC	automatic gain control
AHB	advanced high-performance bus
AI	audio input
AIU	audio input unit
ALU	arithmetic logic unit
AMBA	advanced microcontroller bus architecture
AMP	asymmetric multi-processing
ANI	automatic number identification
ANR	automatic noise reduction



AO	audio output
AOU	audio output unit
AP	access point
APB	advanced peripheral bus
API	application programming interface
APLL	analog phase-locked loop
APSK	amplitude phase shift keying
AQTD	alternate queue transfer descriptor
ARM	advanced RISC machines
ARGB	alpha, red, green, blue
ASF	advanced specification format
ATA	advanced technology attachment
ATAH	ATA host controller
ATAPI	advanced technology attachment packet interface
ATR	answer to reset
ATTR	attribute
AUD	audio
AV	audio & video
AVI	auxiliary video information
AVS	audio video coding standard
AWB	automatic white balance
AXI	advanced eXtensible interface

B

BB	baseband
BCH	Bose-Chaudhuri-Hocquenghem
BCM	byte counter modified
BEP	boot entrance point
BER	bit error rate
BGA	ball grid array
BIST	built-in self test
BIU	bus interface unit
BMC	bi-phase mark coding



BND	bayonet nut connector
BOM	bill of material
BPD	bit plan decoder
BPSK	binary phase shift keying
BRG	bridge
BSP	board support package
BVACT	bottom vertical active area
BVBB	bottom vertical back blank
BVFB	bottom vertical front blank

C

CA	conditional access
CABAC	context-based adaptive binary arithmetic coding
CAR	committed access rate
CAS	column address signal.
CAVLC	context adaptive variable length coding
CBC	cipher block chaining
CBR	constant bit rate
CCB	change control board
CCC	command completion coalescing
CCD	charge-coupled device
CCM	constant coding and modulation
CD	command done or collision detection
CDR	clock data recovery
CEC	consumer electronics control
CF	compact flash
CFB	cipher feedback
CFR	crest factor reduction
CGI	common gate interface
CGMS	copy generation management system
CI	common interface
CIC	cascaded integrator comb
CIU	card interface unit



CL	CAS latency
CLK	clock
CML	current mode logic
CMOS	complementary metal-oxide semiconductor
CN	carrier noise
CNG	comfort noise generator
CODEC	coder/decoder
CP	charge pump
CPL	completion
CPLD	complex programmable logic device
CPU	central processing unit
CR	carrier recovery
CRAMFS	compressed ROM file system
CRC	cyclic redundancy check
CRG	clock and reset generator
CRS	completion retry request
CS	chip select
CSA	common scramble algorithm
CSI	camera serial interface
CSIX	common switch interface
CSMD	carrier sense multiple access
CTI	chroma transient improvement
CTR	counter
CTS	clear to send
CVBS	composite video broadcast signal
CW	cipher word

D

DAC	digital-to-analog converter
DAG	digital automatic gain
DAGC	digital automatic gain control
DAV	DMA of audio and video
DC	direct current



DCD	data connect detection
DCRC	data CRC error
DDC	display data channel
DDR	double data-rate
DDRC	double data rate controller
DHCP	dynamic host configuration protocol
DEM	dynamic-element matching
DES	data encryption standard
DFT	design for testability
DIP	dual in-line package
DIS	digital image stabilization
DiSEqC	digital satellite equipment control
DLL	delay locked loop
DM	data mask
DMA	direct memory access
DMAC	direct memory access controller
DNR	digital noise reduction
DP	data path
DPLL	digital phase-locked loop
DQ	data input/output
DQS	data strobe
DR	design requirement
DRAM	dynamic random access memory
DRC	dynamic range compression
DRM	digital rights management
DRTO	data read timeout
DSI	display serial interface
DSU	dedicated scaling unit
DTMF	dual tone multi frequency
DTO	data transfer over
DVB	digital video broadcasting
DVB-S	digital video broadcasting-satellite
DVD	digital versatile disc



DVI	digital visual interface
DVR	digital video recorder
DWA	data weighted averaging
E	
E2PROM	electrically erasable programmable read-only memory
EAV	end of active video
EB	eviction buffer
EBE	end-bit error
EBI	external bus interface
ECB	electronic codebook
ECC	error correcting code
ECM	entitlement control message
ECS	embedded CPU subsystem
ED	exposed die
EDID	extended display identification data
EEE	energy efficient Ethernet
EHCI	enhanced host controller interface
EMI	electromagnetic interference
EMM	entitlement management message
eMMC	embedded multimedia card
EOP	end of PES
EoS	Ethernet over SONET/SDH
EP	end point
EPG	electronic program guide
EQU	equalizer
ERR	error
ES	element stream
eSATA	external serial advanced technology attachment
ESD	electrostatic discharge
ESR	equivalent series resistance
ETH	Ethernet
ETU	elementary time unit



F

FAS	frame aligning signal
FBE	feedback equalizer
FC	switch fabric
FCBGA	flip-chip ball grid array
FCCSP	flip-chip chip scale package
FEC	forward error correction
FER	frame error rate
FFC	flexible flat cable
FFE	feed forward equalizer
FIFO	first in first out
FIQ	fast interrupt request
FIR	finite impulse response
FIS	frame information structure
FOD	field order detect
FPC	flexible printed connector
FPU	floating-point unit
FRUN	FIFO underrun/overrun error
FSK	frequency shift keying
FTP	File Transfer Protocol

G

GFP-F	frame-mapped generic framing procedure
GFP-T	transparent generic framing procedure
GHB	global history buffer
GIC	generic interrupt controller
GOP	group of picture
GS	generic stream
GMAC	gigabit media access control
GND	ground
GPIO	general purpose input/output
GPL	GNU general public license



GPU graphics processing unit

H

HBA	host bus adapter
HBP	horizontal back porch
HD	high definition
HDCP	high-bandwidth digital content protection
HDI	high density interconnector
HDMI	high definition multimedia interface
HFP	horizontal front porch
HIAO	high-performance audio output interface
HPW	horizontal pulse width
HSTL	high speed transceiver logic
HTML	hypertext markup language
HACT	horizontal active area
HFB	horizontal front blank
HL	high level
HLDC	horizontal lens distortion correction
HLE	hardware locked error
HNP	host negotiation protocol
HTO	data starvation-by-host timeout
HP	high profile
HSIC	high-speed inter-chip
HSS	high-speed serializer/deserializer
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HVBB	horizontal back blank

I

I	in-phase
IBIS	input/output buffer information specification
IC	integrated circuit
I²C	inter-integrated circuit



I²S	inter-IC sound
I/O	input/output
IOC	I/O configuration
IP	Internet Protocol
ISI	input stream identifier
ISP	image signal processor
IDE	integrated device electronic
LDPC	low density parity check code
IDR	intermediate data rate
IF	intermediate frequency
IGMP	Internet Group Management Protocol
LMS	linear mean square
IPF	IP filter
IPv4	Internet Protocol Version 4
IR	infrared
IRQ	interrupt request
ISI	input stream identifier
ISP	image signal processor
ISR	interrupt service routine
ITCM	instruction tightly coupled memory
ITLA	integrated tunable laser assembly
ITU	International Telecommunication Union
IV	initialization vector

J

JFFS2	journaling flash file system version 2
JPEG	Joint Photographic Experts Group
JPGE	JPEG encoder
JTAG	Joint Test Action Group

K

KL	key ladder
-----------	------------



L

LCD	liquid crystal display
LDO	low dropout regulator
LDPC	low-density parity check code
LED	light emitting diode
LFB	line fill buffer
LFSR	linear feedback shifting register
LMR	load mode register
LMS	least mean square
LNB	low noise block
LOS	loss of signal
LPI	low-power idle
LRB	line read buffer
LSB	least significant bit
LSP	label switched path
LSN	logic sector number
LTI	luma transient improvement
LVDS	low-voltage differential signaling
LVPECL	low-voltage positive emitter coupled logic
LVTTL	low-voltage transistor-transistor logic
LVPECL	low-voltage positive emitter-coupled logic

M

MAC	media access control
MBAFF	macroblock adaptive frame field
MCE	media control engine
MCU	microprogrammed control unit
MD	motion detection
MDDRC	multiport DDRC
MDIO	management data input/output
MDU	motion detect unit
MF	matched filter
MQFN	mapped quad flat non-leaded



MHL	mobile high-definition link
MII	media independent interface
MIPI	mobile industry processor interface
MIPS	microprocessor without interlocked pipeline stages
MLC	multi-level cell
MLF	malformed
MMB	media memory block
MMC	multimedia card
MMU	memory management unit
MMZ	media memory zone
MP	main profile
MPI	MPP programming interface
MPE	media processing engine
MPLL	multiplying phase-locked loop
MPP	media processing platform
MRL	manually-operated retention latch
MSB	most significant bit
MSE	mean square error
MSG	message
MV	motion vector

N

NAL	network abstraction layer
NANDC	NAND flash controller
NC	not connect
NCQ	native command queuing
NLP	non-linear processor
NR	noise reduction
NRZ	non-return-to-zero
NTSC	National Television Systems Committee
NVR	network video recorder

O



OCT	on-chip termination
OD	open drain
ODT	on-die termination
OEN	output enable
OFB	output feedback
OHCI	open host controller interface
OOB	out of band
OP	operational amplifier
OR	original requirement
OSC	oscillator
OSD	on screen display
OTG	on-the-go
OTP	one time programmable
OTU	optical transponder unit

P

PAD	packet assembler/disassembler
PAFF	picture adaptive frame field
PAL	phase alternating line
PCB	printed circuit board
PCI	peripheral component interconnect
PCIe	peripheral component interconnect express
PCIV	PCI view
PCR	program clock reference
PCM	pulse code modulation
PDM	pulse density modulation
PECL	positive emitter coupled logic
PER	packet error rate
PES	packetized elementary stream
PG	power/ground
PHY	physical
PID	packet ID
PIM-DM	protocol independent multicast dense mode



PIM-SM	protocol independent multicast sparse mode
PIO	programmable input/output
SSA	secure software authentication
PLL	phase-locked loop
PLS	physical layer signaling
PM	port multiplexer
PMoC	power management of chip
PMP	personal media player
POR	power-on reset
PPP	Point-to-Point Protocol
PPS	picture parameter set
PRBS	pseudo random binary sequence
PRDT	physical region descriptor table
PSI	program specific information
PSK	phase shift keying
PSRAM	pseudo static random access memory
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
PT	packet type
PTS	presentation time stamp
PUB	PHY utility block
PUSI	payload unit start indicator
PWM	pulse width modulation

Q

Q	quadrant
QAM	quadrature amplitude modulation
QDR	quad data rate
QoS	quality of service
QP	quantizer parameter
QPSK	quaternary phase shift keying

R



RAM	random access memory
RAS	row address signal
RC	resistor-capacitor
RCA	Radio Corporation of America
RCRC	response CRC error
RE	response error
RF	radio frequency
RGB	red-green-blue
RGMII	reduced gigabit media independent interface
RH	relative humidity
RoHS	restriction of the use of certain hazardous substances
ROI	region of interest
ROM	read-only memory
ROP	raster operation
RPR	resilient packet ring
RLDRAM	reduced latency dynamic random access memory
RMII	reduced media-independent interface
RS	Reed-Solomon
RTC	real-time clock
RTO	response timeout
RTS	request to send
RVDS	RealView development suite
RX	receive
RXDR	receive FIFO data request

S

SAP	service access point
SAD	sum of absolute difference
SAR	successive approximation
SATA	serial advanced technology attachment
SAV	start of active video
SBE	start-bit error
SBP	secure boot procedure



SCD	start code detect
SCI	smart card interface
SCL	serial clock
SCR	system clock reference
SCS	secure chipset start-up
SCU	snoop control unit
SD	secure digital
SDA	serial data
SDB	set device bits
SDH	synchronous digital hierarchy
SDHC	secure digital high capacity
SDI	serial digital interface
SDIO	secure digital input/output
SDK	software development kit
SDRAM	synchronous dynamic random access memory
SDV	system design verification
SI	specific information
SIO	sonic input/output
SLC	single-level cell
SMI	static memory interface
SNAP	subnetwork access point
SNR	signal-to-noise ratio
SNTF	serial ATA notification
SOA	semiconductor optical amplifier
SoC	system-on-chip
SONET	synchronous optical network
SOP	start of PES
SP	simple profile
SPDIF	Sony/Philips digital interface
SPI	serial peripheral interface
SPS	sequence parameter set
SRAM	static random access memory
SRP	Session Request Protocol



SSA	secure software authentication
SSD	secure software download
SSMC	synchronous static memory controller
SSP	synchronous serial port
SSRAM	synchronous static random access memory
SSTL-18	stub series terminated logic for 1.8 V
STA	station
STB	set-top box
STM-1	synchronous transport module level 1
SVB	selective voltage bing
SYNC	synchronization
SYS	system

T

TBD	to be determined
TBGA	tape ball grid array
TC	traffic class
TCP	Transmission Control Protocol
TD	TLP digest
TDES	triple data encryption standard
TDE	two-dimensional engine
TE	tearing effect
TEI	transport error indicator
TFD	task file data
TFPBGA	tape fine-pitch ball grid array
TFT	thin-film technology
TI	Texas Instruments
TLV	type-length-value
TOE	TCP/IP offload engine
TP	transponder
TPIT	TS packet index table
TR	timing recovery
TS	transport stream



TSI	transport stream interface
TT	teletext
TV	television
TVACT	top vertical active area
TVBB	top vertical back blank
TVFB	top vertical front blank
TVS	transient voltage suppressor
TX	transmit
TXDR	transmit FIFO data request

U

UART	universal asynchronous receiver transmitter
U-boot	universal boot loader
UC	unexpected completion
UDP	User Datagram Protocol
ULPI	UTMI low pin interface
UPnP	universal plug and play
UR	unsupported request
USB	universal serial bus
USIM	universal subscriber identity module
UTMI	USB 2.0 transceiver macrocell interface

V

VACT	vertical active area
VAD	voice activity detector
VAPU	video analysis&process unit
VBB	vertical back blank
VBI	vertical blanking interval
VBR	variable bit rate
VCC	common connector voltage
VCO	voltage controller oscillator
VCM	variable coding and modulation
VCMP	video compress



VCXO	voltage control crystal oscillator
VDA	video detection analysis
VDH	video decoder for high-definition
VDM	video decoding module
VDEC	video decoding
VDP	video display
VEDU	video encoding/decoding unit
VENC	video encoding
VFB	vertical front blank
VFMW	video firmware
VFP	vertical front porch
VGA	video graphics array
VI	video input
VIC	vector interrupt controller
VICAP	video capture
VIU	video input unit
VLD	valid
VLL	virtual leased line
VO	video output
VOIE	voice encoder
VOU	video output unit
VPP	video pre-processing
VPS	video programming system
VPSS	video process subsystem
VPW	vertical pulse width
VSA	vertical sync start
VQE	voice quality enhancement
VQM	voice quality monitor

W

WDG	watchdog
WE	write enable
WFE	wait for event



WFI	wait for interrupt
WRED	weighted random early discard
WSS	wide screen signaling

X

XAUI	10 gigabit attachment unit interface
-------------	--------------------------------------

Y

YAFFS	yet another flash file system
YUV	luminance-bandwidth-chrominance

Z

ZME	zoom engine
------------	-------------



Contents

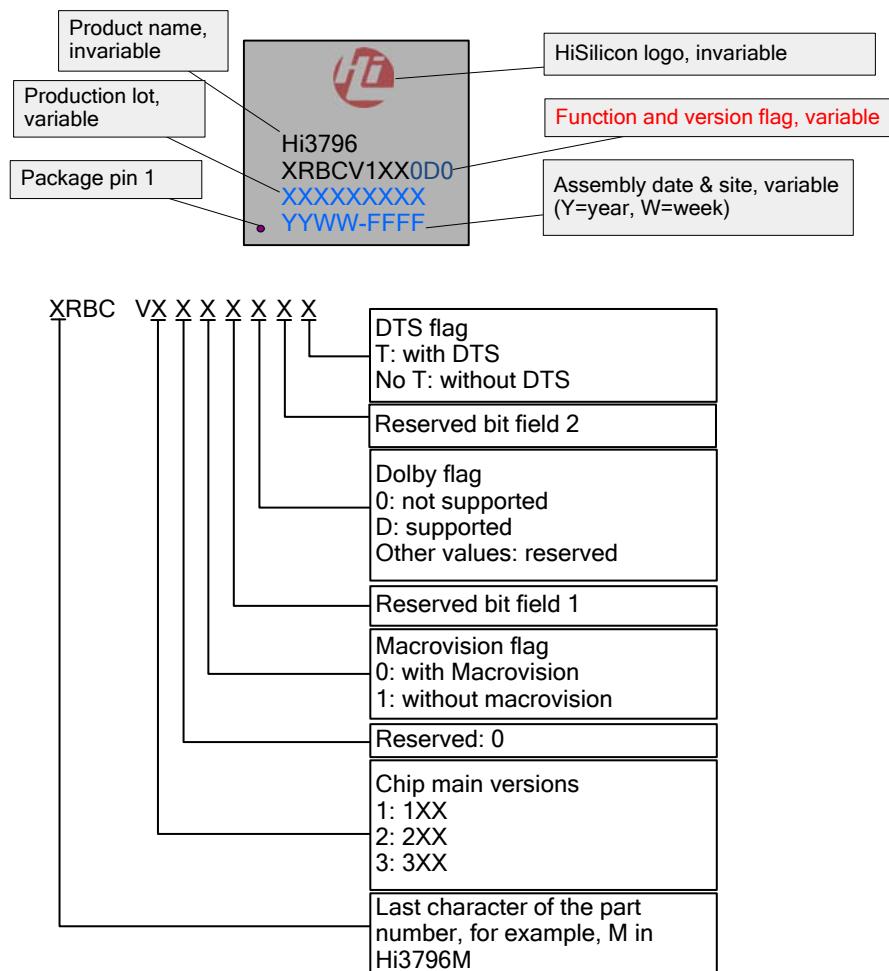
B Ordering Information.....	B-1
------------------------------------	------------



B Ordering Information

Figure B-1 illustrates the mark naming convention of Hi3796M V100.

Figure B-1 Mark naming convention



HiXXXX indicates the 6-digit product model before the part number.

The letter X in the part number indicates variable digits and is only for internal use.



Table B-1 Packages

Part Number	Package	Body Size	Pitch
Hi3796M RBC V10X XXX	PBGA 573	23 mm x 23 mm (0.91 in. x 0.91 in.)	0.8 mm (0.031 in.)

For example, if the part number is Hi3796M RBCV1010D0, it indicates that Hi3796M V100 with the BGA package supports Dolby but not Macrovision (Rovi) and DTS.



CAUTION

According to the technical agreement between HiSilicon and providers of third-party software, HiSilicon has no rights to provide technical samples of the third-party software for other parties. Therefore, if the third-party software or chips that are provided for only authenticated users are involved (including but not limited to DD+ and DTS), ensure that you have obtained authentication from the third party when you place an order for the HiSilicon chip.