Hi3861 V100/Hi3861L V100/Hi3881 V100 Wi-Fi Chip

Data Sheet

Issue 02

Date 2020-07-29

Trademarks and Permissions

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

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

Notice

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

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

HiSilicon (Shanghai) Technologies Co., Ltd.

Address: New R&D Center, 49 Wuhe Road, Bantian,
Longgang District,
Shenzhen 518129 P. R. China

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

Email: support@hisilicon.com

Contents

Figures

Tables

Contents

About This Document

Purpose

This document describes the basic functions and configuration methods in common application modes of Hi3861 V100, Hi3861L V100, and Hi3881 V100.

This document also describes how to configure Hi3861 V100, Hi3861L V100, and Hi3881 V100.

Related Versions

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

Product Name	Version
Hi3861	V100
Hi3861L	V100
Hi3881	V100

Intended Audience

This document is intended for:

- Technical support engineers
- R&D engineers of customers

Symbol Conventions

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

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

Symbol	Description
WARNING	Indicates a hazard with a medium level of risk which, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazard with a low level of risk which, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
NOTE	Supplements the important information in the main text. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Register Attributes

Type	Description	Type	Description
RO	The register is read-only.	RW	The register is readable and writable.
RC	The register is cleared when it is read.	WC	This register is readable. Writing **1** clears the register, and writing **0** remains unchanged.

Reset Value Conventions

In the register definition tables:

- If the reset value (in the reset row) of a bit is **?**, the reset value is uncertain.
- If one or more bits are marked with **?**, the reset value of the entire register (**Total Reset Value**) is **-**, indicating that the reset value is uncertain.

Change History

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

Issue 02 (2020-07-29)

Descriptions and precautions about incremental changes to the chip in section 6.10.2 "Function Description" are deleted.

Section 6.10.3 "Sleep Wakeup Control" is deleted.

Section 6.10.4 "Address Remapping" is deleted.

Section 6.10.5 "Common Access Registers" is deleted.

Issue 01 (2020-04-30)

This issue is the first official release.

1 Introduction

1.1 Overview

Hi3861/Hi3881 is a highly integrated 2.4 GHz Wi-Fi chip that packs the IEEE 802.11 b/g/n baseband and radio frequency (RF) circuit. The RF circuit consists of the power amplifier (PA), low-noise amplifier (LNA), RF balun, antenna switch, and power management module.

The Wi-Fi baseband of Hi3861/Hi3881 implements the orthogonal frequency division multiplexing (OFDM) technology and is backward compatible with the direct sequence spread spectrum (DSSS) and complementary code keying (CCK) technologies. IEEE 802.11 b/g/n standards are complied. It operates over a 20 MHz standard bandwidth or a 5 MHz/10 MHz narrow bandwidth, at a maximum physical layer bit rate of 72.2 Mbit/s.

Hi3861/Hi3881 integrates a high-performance 32-bit microprocessor and offers various peripheral interfaces such as serial peripheral interface (SPI), universal asynchronous receiver transmitter (UART), inter-integrated circuit (I^2C), inter-IC sound (I^2S), pulse-width modulation (PWM), general purpose input/output (GPIO), and multi-channel analog-to-digital converter (ADC), as well as the secure digital input and output (SDIO) 2.0 interface, with clock frequencies up to 50 MHz. Huawei LiteOS and third-party components are supported. In addition, an open and easy-to-use development and debugging environment is provided.

The Hi3861 series include Hi3861 V100 and Hi3861L V100. Its built-in static random access memory (SRAM) and flash can operate independently. Programming is allowed on the flash.

The Hi3881 series includes Hi3881 V100. As a slave chip, the chip runs by connecting to the master chip over the SDIO interface.

1.2 Features

Hi3861/Hi3881 has the following features:

- General specifications
  - 1 x 1 2.4 GHz frequency band (channels 1–14)
  - Physical layer (PHY) supporting IEEE 802.11 b/g/n
  - Media Access Control (MAC) supporting IEEE 802.11 d/e/h/i/k/v/w
  - Built-in PA and LNA, integrated with TX/RX switch and balun
  - Station (STA) and access point (AP) modes, up to six STA devices allowed for access as an AP

- WFA WPA, WFA WPA2 personal, and WPS2.0
- 2/3/4-line packet traffic arbitration (PTA) solution with Bluetooth (BT) and Bluetooth low energy (BLE) chips coexisting
- RF auto-calibration solution
- Input voltage range: 2.3–3.6 V
- I/O supply voltage: 1.8 V or 3.3 V
- Hi3861L V100 power consumption:

  Tested at the ambient temperature of 25 ℃:

  Ultra deep sleep mode: 3 μA@3.3 V

  Tested in a shielded environment at the ambient temperature of 25 ℃, with the RX duration of 1 ms, and powered by the buck:

  DTIM1: 0.97 mA@3.6 V

  DTIM3: 0.36 mA@3.6 V

  DTIM10: 0.15 mA@3.6 V
- Hi3861 V100 power consumption:

  Tested at the ambient temperature of 25 ℃:

  Ultra deep sleep mode: 3 μA@3.3 V

  Tested in a shielded environment at the ambient temperature of 25 ℃, with the RX duration of 1 ms, and powered by the buck:

  DTIM1: 1.27 mA@3.6 V

  DTIM3: 0.523 mA@3.6 V

  DTIM10: 0.233 mA@3.6 V

- PHY specifications
  - All single-antenna data rates of IEEE 802.11b/g/n
  - Maximum rate of 72.2 Mbit/s at HT20 MCS7
  - Standard 20 MHz bandwidth and 5 MHz/10 MHz narrow bandwidth
  - Space-Time Block Coding (STBC) RX
  - Short guard interval (GI)
- MAC specifications
  - Aggregate MAC protocol data unit (A-MPDU)
  - Block-Acknowledgment (Blk-ACK)
  - Quality of service (QoS)
- CPU specifications
  - High-performance self-developed 32-bit CPU with a maximum operating frequency of 160 MHz
  - Built-in 352 KB SRAM and 288 KB read-only memory (ROM)
  - Built-in 2 MB flash (Hi3861 only)
  - Peripheral interfaces (implemented by multiplexing)

    Hi3861: one SDIO slave interface, two SPI interfaces, two I^2C interfaces, three UART interfaces, 15 GPIO interfaces, seven ADC inputs, six PWM interfaces, one I^2S interface, and a 32 kHz external clock (Hi3861L V100 only)

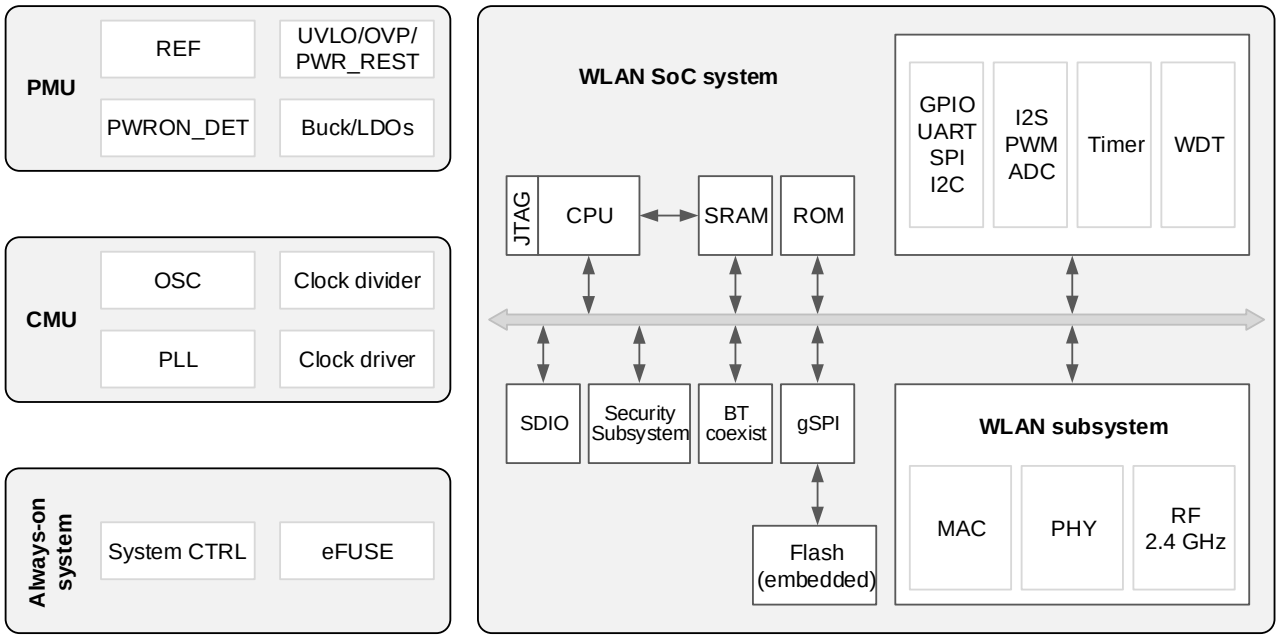
    Hi3881: one SDIO slave interface, one UART interface, and five GPIO interfaces
- Other Information

- Package information: 5 mm x 5 mm QFN-32
- Operating temperature: –40 ℃ to +85 ℃ (–40 ℉ to 185 ℉)

1.3 Logic Block Diagram

Figure 1-1 shows the logic block diagram of Hi3861.

Figure 1-1 Logical block diagram of Hi3861



Note: PMU is short for the power management unit. REF is short for the reference indicator. UVLO is short for under voltage lock out. OVP is short for overvoltage protection. LDO is short for the low dropout regulator. CMU is short for the clock managing unit. OSC is short for the Oscillator. PLL is short for the phase-locked loop. eFUSE is short for the electrical FUSE. JTAG is short for the joint test action group. WDT is short for the watch dog timer.

Figure 1-2 shows the logic block diagram of Hi3881.

Figure 1-2 Logical block diagram of Hi3881

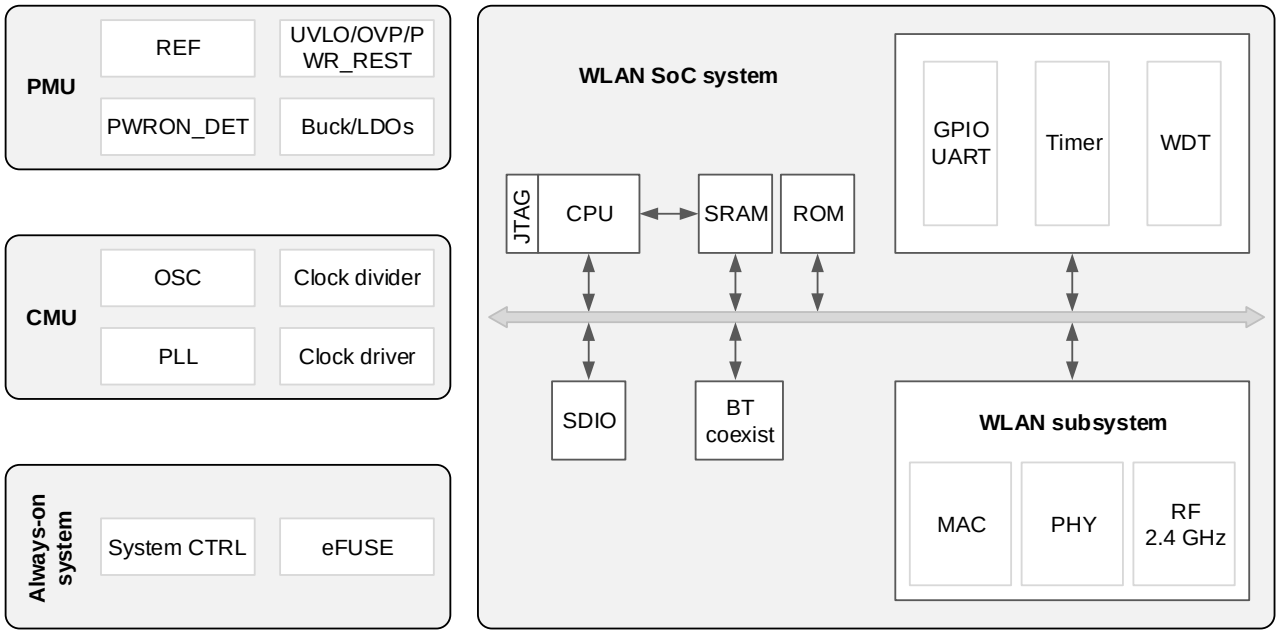


Table 1-1 describes the functions of each module in the logic block diagram.

Table 1-1 Module description

Module Name	Description
PMU	Power management unit
CMU	Clock managing unit
GPIO	General-purpose input/output
UART	Universal asynchronous serial interface controller
SPI	Serial peripheral interface
I^2C	Bidirectional two-wire synchronous serial bus
I^2S	Audio bus
ADC	Analog to digital converter
PWM	Pulse signal width modulation
Timer	Timer
WDT	Watchdog timer
Security subsystem	Security subsystem
SRAM	Static random access memory
eFUSE	Data encryption and decryption and chip ID storage
MAC	Service processing at the MAC layer

Module Name	Description
PHY	Channel modulation and demodulation
RF	RF module

1.4 Typical Applications

Hi3861 V100 is applicable to IoT smart terminals such as smart home appliances.

Hi3861L V100 is applicable to IoT low-power intelligent products such as smart home appliances, intelligent lock, low-power cameras, and switch buttons.

Hi3881 V100 is applicable to the IP camera, over the top (OTT), set-top box (STB), and TV appliances.

2 System

2.1 Reset

2.1.1 Overview

The reset module generates a reset signal for each module based on the input control signal. It supports global reset of the entire chip and separate reset of a single module as well.

A reset is asserted asynchronously and deasserted synchronously.

2.1.2 Reset Control

Reset control require the following input signals:

- Global reset control
- Single-module reset control

Table 2-1 describes the reset control modes supported by the chip.

Table 2-1 Reset control

Reset Mode	Source	Reset Time	Reset Range	Chip Mode After Reset
Power-on reset 11 (POR11)	1.1 V POR module output	200 μs	Entire chip, with the hardware control word and JTAG enable signal latched	Work
WDT reset control	WDT module output, which can be masked by software	-	Module	Work
Global soft reset control	Software configuration	-	Entire chip	Work
Subsystem soft reset control	Software configuration	-	CPU subsystem	Work

Reset Mode	Source	Reset Time	Reset Range	Chip Mode After Reset
Module-level soft reset control	Clock and reset generator (CRG) register, configured by software	Software configuration	Module	Hold
Module-level hard reset control	Chip module, masked by software	-	Module	Hold

2.2 Clock Management

2.2.1 Overview

The clock management module manages the chip clock input, generation, and control in a unified manner. The clock management module provides the following features:

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

2.2.2 Clock Allocation

Table 2-2 Clock allocation in normal operating

Module	Clock Frequency (MHz)	Module	Clock Frequency (MHz)
Huawei-developed CPU	160	Security subsystem (SSS)	160
CPU BUS	160	GPIO	Crystal/0.032
TOP BUS	80	Serial peripheral interface flash controller (SFC)	96 (adjustable, depending on the external component)
Timer	Crystal frequency division	UART	160/80
Real-time clock (RTC)	Crystal/0.032	I^2C	80
Watchdog	80	SPI controller	160

PWM	160/crystal	EFUSE	Crystal
MAC	80	PMU OSC	0.032
PHY	160/80/40/20	Crystal	40/24
ADC	80	Direct memory access (DMA)	80
Digital analog converter (DAC)	160	SDIO	50
AUDIO_AIAO	12.288	-	-

2.2.3 Clock Control

The clock management module performs the following operations:

- Controls the PLL frequency
- Controls the clock division and clock source selection.
- Manages clock gating

2.3 Power Management and Low-Power Mode Control

2.3.1 Overview

In low-power mode, the power consumption of the chip is reduced effectively. The chip dynamically reduces its power consumption in the following low-power control modes:

- System operating mode control

  In each operating mode except the work 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.

NOTICE

- Hi3861 V100 supports all modes. It has no 32 kHz external clock.
- Hi3861L V100 supports all modes. It has the 32 kHz external clock.
- Hi3881 V100 doe not support deep sleep or ultra-deep sleep. It has no 32 kHz external clock.

2.3.2 System Operating Modes

The system can work in the following modes:

- Work

  In work (normal working) mode, all power supplies are turned on to support Wi-Fi services.

- Light Sleep

  In light sleep mode, service TX and RX can be quickly resumed, the TX and RX clocks are disabled to reduce power consumption. In this mode, the CPU is to the WFI mode. After the CPU is woken up by interrupt, service TX and RX are resumed. The flash and I/O supply (VDDIO) remains alive. The software determines whether to power down the SDIO. The system can be woken up by using the SDIO interrupt. The SoC clock source can be selected between the crystal or PLL, whose frequency is related to the communication rate of peripherals.

- Deep sleep

  In deep sleep mode, most services are powered off to reduce power consumption. The GPIO, RTC, and external timing synchronization function (TSF) modules are reserved and can be woken up from deep sleep. The system is powered up again and waits for the CPU to start service TX and RX. In this case, the flash memory is powered down, and I/O supply remains alive. In this mode, SDIO power supply can be reserved by configuring the software, so that the system can be woken up by using the SDIO (the SDIO is not in sleep mode). RAM can be set to retention.

- Ultra-deep sleep

  In ultra-deep sleep mode, the chip consumes the least power consumption. Most power supplies of the chip are turned off to reduce power consumption, including the RAM supply. The system can be woken up by using pin GPIO3/5/7/14.

- Shutdown

  The chip enters the shutdown mode once the PMU_PWRON pin is pulled down. When the PMU_PWRON pin is pulled up, the system exits the shutdown mode and enters the work mode.

2.4 CPU Subsystem

The system provides a Huawei-developed processor as the main control CPU to complete system tasks and control tasks. The CPU has a 32 KB I-cache and a 4 KB D-cache.

The CPU has the following features:

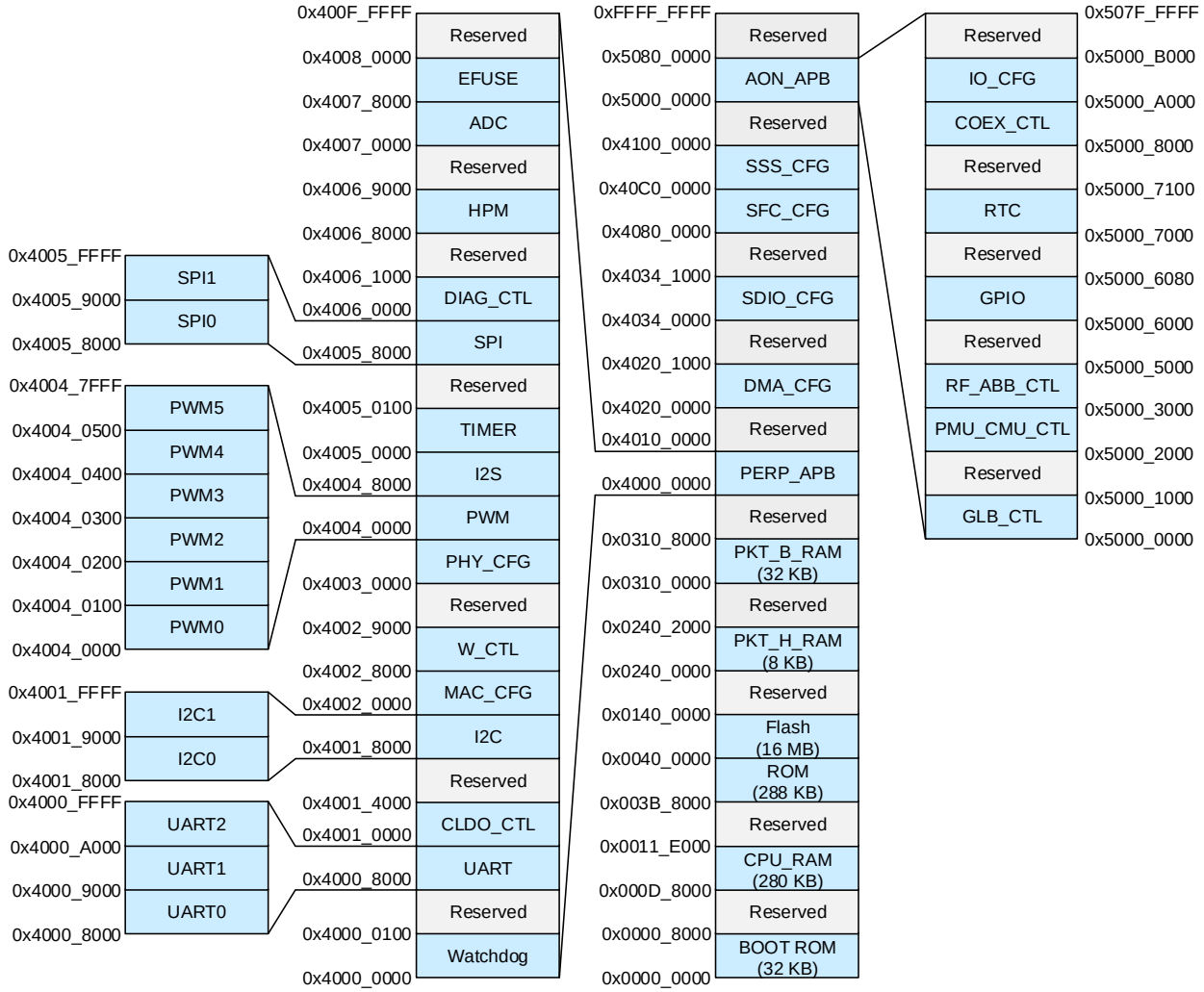
- Up to 160 MHz operating frequency
- Direct interrupt mode and vectored interrupt mode, supporting 35 non-standard external interrupts

- Edge-triggered interrupts and level-triggered interrupts
- Flash patching, with 254 instruction comparators and 2 address comparators
- Physical memory protection (PMP)
- JTAG and Serial Wire Debug (SWD) interfaces for debug use

2.5 Memory Space Mapping

Figure 2-1 shows the address mapping for remapping.

Figure 2-1 Address mapping for remapping



Note: HPM is short for the hardware performance monitor.

2.6 System Tick

2.6.1 Overview

System Tick is a timer of the system. The timer time varies according to the clock frequency.

2.6.2 Features

System Tick has the following features:

- By default, the 64-bit System Tick timer ticks clockwise by using the 32 kHz clock generated by the POR to keep the system time.
- The timer can be cleared by software. The software can lock the timer to read the timer value or read the timer value in real time.
- The timer can be cleared by POR, chip global reset, and software reset only.

2.7 Interrupt System

2.7.1 Interrupt Allocation

Hi3861, Hi3861L, and Hi3881 support the vectored interrupts and direct interrupts in edge-triggered and level-triggered modes. The chips support eight programmable priorities by using the priority configuration register (3 bits in total).

The interrupt system includes:

- Internal standard interrupts of the CPU, interrupts 0–25.
- Non-standard interrupts outside the CPU, as described in Table 2-3.

Table 2-3 Non-standard interrupts

Interrupt No.	Interrupt	Interrupt No.	Interrupt
26	Timer0 interrupt	44	SPI1 interrupt
27	Timer1 interrupt	45	GPIO interrupt
28	Timer2 interrupt	46	T-Sensor interrupt
29	RTC0 interrupt	47	wlan_sleep interrupt
30	RTC1 interrupt	48	wlan_wakeup interrupt
31	RTC2 interrupt	49	over_temp interrupt
32	RTC3 interrupt	50	pmu_cmu_err interrupt
33	WDT interrupt	51	Soft interrupt 0
34	Mac interrupt	52	Soft interrupt 1
35	DMA interrupt	53	Soft interrupt 2
36	SFC interrupt	54	Soft interrupt 3

Interrupt No.	Interrupt	Interrupt No.	Interrupt
37	SDIO interrupt	55	PEK interrupt
38	UART0 interrupt	56	Advanced Encryption Standard (AES)/Secure Hash Algorithm 256 (SHA256)/key derivation function (KDF) interrupt
39	UART1 interrupt	57	TRNG interrupt
40	UART2 interrupt	58	LSADC interrupt
41	I2C0 interrupt	59	I2S0 interrupt
42	I2C1 interrupt	60	Ultra-deep sleep interrupt
43	SPI0 interrupt	-	-

Note: The machine timer interrupt of the CPU uses the timer3 interrupt source. Timer3 cannot be used for other purposes.

2.7.2 Interrupt Structure

NOTE

- Hi3861, Hi3861L, and Hi3881 use the interrupt controller integrated in the CPU. All the non-standard interrupts triggered by peripherals or registers are directly connected to the CPU for processing.
- This section describes non-IP interrupts only. For details about the interrupts of each IP, see the corresponding section.

2.7.2.1 NMI Interrupts

The non-maskable interrupt (NMI) is used to control software and logic to burn data to specified area of the eFUSE. It is triggered by the TEE_NMI_INT register ([tee_nmi_int] = 1).

NOTICE

TEE_NMI_INT[tee_nmi_int] is not automatically cleared. After the interrupt is triggered, this bit needs to be written back to 0. After the NMI interrupt is triggered, the CPU notifies the eFUSE control logic that the CPU is in NMI interrupt mode. An area in eFUSE can be burnt only in this mode. For details, see section 5.7 "eFUSE."

2.7.2.2 Software Interrupt

Four software interrupts (0–3) are provided which are triggered by configuring registers.

The software interrupts are triggered in the same way. The following describes how to trigger soft interrupt 0.

Step 1 Write **1** to SOFT_INT_EN[soft_int0_en] to enable soft interrupt 0.

Step 2 Write **1** to SOFT_INT_SET[soft_int0_set] to set software interrupt 0.

This register is cleared to 0 automatically. Writing **1** automatically clears the register. Writing **0** takes no effect.

Step 3 Software starts software interrupt 0 processing. In this case, SOFT_INT_STS[soft_int0_sts] can be read to query the interrupt status.

- 0: no interrupt
- 1: interrupt generated

Step 4 Write **1** to SOFT_INT_CLR[soft_int0_clr] to clear software interrupt 0.

This register is cleared to 0 automatically. Writing **1** automatically clears the register. Writing **0** takes no effect.

----End

2.7.3 Register Summary

Table 2-4 describes the interrupt system registers.

Table 2-4 Summary of interrupt system registers

Absolute Address	Register	Description	Page
0x40010258	TEE_NMI_INT	NMI interrupt control register	2-8
0x50000280	SOFT_INT_EN	Software interrupt enable register	2-9
0x50000284	SOFT_INT_SET	Software interrupt triggering register	2-9
0x50000288	SOFT_INT_CLR	Software interrupt clear register	2-10
0x5000028C	SOFT_INT_STS	Software interrupt status query register	2-10

2.7.4 Register Description

TEE_NMI_INT

TEE_NMI_INT is the NMI interrupt control register.

Absolute Address: 0x40010258 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:1]	RO	reserved	reserved	0x0000
[0]	RW	tee_nmi_int	NMI interrupt control 0: pulled down 1: pulled up	0x0

SOFT_INT_EN

SOFT_INT_EN is the software interrupt enable register.

Absolute Address: 0x50000280 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	RO	reserved	Reserved	0x000
[3]	RW	soft_int3_en	Software interrupt 3 enable 0: disabled 1: enabled	0x0
[2]	RW	soft_int2_en	Software interrupt 2 enable 0: disabled 1: enabled	0x0
[1]	RW	soft_int1_en	Software interrupt 1 enable 0: disabled 1: enabled	0x0
[0]	RW	soft_int0_en	Software interrupt 0 enable 0: disabled 1: enabled	0x0

SOFT_INT_SET

SOFT_INT_SET is the software interrupt triggering register.

Absolute Address: 0x50000284 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	RO	reserved	Reserved	0x000
[3]	W1_PULSE	soft_int3_set	Soft interrupt 3 triggering 0: invalid 1: triggered	0x0
[2]	W1_PULSE	soft_int2_set	Soft interrupt 2 triggering 0: invalid 1: triggered	0x0
[1]	W1_PULSE	soft_int1_set	Soft interrupt 1 triggering 0: invalid 1: triggered	0x0
[0]	W1_PULSE	soft_int0_set	Soft interrupt 0 triggering 0: invalid	0x0

			1: triggered	

SOFT_INT_CLR

SOFT_INT_CLR is the software interrupt clear register.

Absolute Address: 0x50000288 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	RO	reserved	Reserved	0x000
[3]	W1_PULSE	soft_int3_clr	Soft interrupt 3 clear 0: invalid 1:cleared	0x0
[2]	W1_PULSE	soft_int2_clr	Soft interrupt 2 clear 0: invalid 1:cleared	0x0
[1]	W1_PULSE	soft_int1_clr	Soft interrupt 1 clear 0: invalid 1:cleared	0x0
[0]	W1_PULSE	soft_int0_clr	Soft interrupt 0 clear 0: invalid 1:cleared	0x0

SOFT_INT_STS

SOFT_INT_STS is the software interrupt status query register.

Absolute Address: 0x5000028C Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:4]	RO	reserved	Reserved	0x000
[3]	RO	soft_int3_sts	Status of software interrupt 3 0: no interrupt 1: interrupt generated	0x0
[2]	RO	soft_int2_sts	Status of software interrupt 2 0: no interrupt 1: interrupt generated	0x0
[1]	RO	soft_int1_sts	Status of software interrupt 1 0: no interrupt	0x0

			1: interrupt generated	
[0]	RO	soft_int0_sts	Status of software interrupt 0 0: no interrupt 1: interrupt generated	0x0

2.8 RTC

2.8.1 Overview

The RTC counts down from the preset data to 0, triggering an interrupt. Four identical RTC units are provided in the chip. The initial data, working mode, and interrupt mode of each RTC unit can be configured by using registers. The RTC units can be enabled and configured separately. Their clock intervals can also be flexibly specified.

2.8.2 Features

The RTC has the following features:

- Four 32-bit RTC units that can be separately configured are provided.
- Free mode and periodic mode are supported.
- Each RTC unit can be enabled separately.
- Each RTC unit is configured with an independent interrupt.
- The current timer value can be locked.

2.8.3 Operating Modes

The RTC working clock can be selected between the 32 kHz clock and the 40 MHz or 24 MHz crystal clock.

The RTC runs in free mode or periodic mode. Each RTC unit can be configured separately.

- In free mode, a timer counts down from **0xFFFF_FFFF** to **0**.
- In periodic mode, a timer counts down from the value of **TIMER*x*_LOADCOUNT** (*x* = 1–4) to **0**.

The following describes the workflow of the RTC by taking RTC3 and RTC4 as examples:

Step 1 Configure TIMER3_CONTROLREG and TIMER4_CONTROLREG to initialize the RTC.

- Write **0** to TIMER3_CONTROLREG[timer3_en] and TIMER4_CONTROLREG[timer4_en] to disable the RTC.

  Note: To avoid absence of synchronization, RTC must be disabled before setting TIMER3_LOADCOUNT and TIMER4_LOADCOUNT.
- Write **0** to TIMER3_CONTROLREG[timer3_mode] and TIMER4_CONTROLREG[timer4_mode] to set the RTC units to free mode. Or write **1** to set the RTC units to periodic mode.

  Note: Before setting the RTC working mode to free mode, you must set all valid bits of TIMER3_LOADCOUNT and TIMER4_LOADCOUNT to 1.

- Write **1** to TIMER3_CONTROLREG[timer3_int_mask] and TIMER4_CONTROLREG[timer4_int_mask] to enable the interrupt mask. Or write **0** to disable the interrupt mask.

Step 2 Write the initial values of the RTC units to TIMER3_LOADCOUNT and TIMER4_LOADCOUNT (for the periodic mode).

Step 3 Write **1** to TIMER3_CONTROLREG[timer3_en] and TIMER4_CONTROLREG[timer4_en] to enable the RTC units and start the countdown.

----End

2.8.4 Register Summary

Table 2-5 describes the RTC registers.

Table 2-5 Summary of RTC registers (base address: 0x5000_7000)

Offset Address	Register	Description	Page
0x000	TIMER1_LOADCOUNT	Timer1 initial value register	2-13
0x004	TIMER1_CURRENTVALUE	Timer1 current value register	2-13
0x008	TIMER1_CONTROLREG	Timer1 control register	2-13
0x00C	TIMER1_EOI	Timer1 interrupt clear register	2-14
0x010	TIMER1_INTSTATUS	Timer1 interrupt status register	2-14
0x014	TIMER2_LOADCOUNT	Timer2 initial value register	2-14
0x018	TIMER2_CURRENTVALUE	Timer2 current value register	2-15
0x01C	TIMER2_CONTROLREG	Timer2 control register	2-15
0x020	TIMER2_EOI	Timer2 interrupt clear register	2-15
0x024	TIMER2_INTSTATUS	Timer2 interrupt status register	2-16
0x028	TIMER3_LOADCOUNT	Timer3 initial value register	2-16
0x02C	TIMER3_CURRENTVALUE	Timer3 current value register	2-16
0x030	TIMER3_CONTROLREG	Timer3 control register	2-16
0x034	TIMER3_EOI	Timer3 interrupt clear register	2-17
0x038	TIMER3_INTSTATUS	Timer3 interrupt status register	2-17
0x03C	TIMER4_LOADCOUNT	Timer4 initial value register	2-18
0x040	TIMER4_CURRENTVALUE	Timer4 current value register	2-18
0x044	TIMER4_CONTROLREG	Timer4 control register	2-18
0x048	TIMER4_EOI	Timer4 interrupt clear register	2-19
0x04C	TIMER4_INTSTATUS	Timer4 interrupt status register	2-19

Offset Address	Register	Description	Page
0x0A0	TIMERS_INTSTATUS	Interrupt status register for timers	2-19
0x0A4	TIMERS_EOI	Interrupt clear register for timers	2-20
0x0A8	TIMERS_RAWINTSTATUS	Raw interrupt status register	2-20

2.8.5 Register Description

TIMER1_LOADCOUNT

TIMER1_LOADCOUNT is the timer1 initial value register.

Offset Address: 0x000 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer1_loadcount	Initial value of timer1	0x00000000

TIMER1_CURRENTVALUE

TIMER1_CURRENTVALUE is the timer1 current value register.

Offset Address: 0x004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timer1_currentvalue	Current value of timer1	0x00000000

TIMER1_CONTROLREG

TIMER1_CONTROLREG is the timer1 control register.

Offset Address: 0x008 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RW	timer1_lock	Timer1 lock control 0: not locked 1: Locks the timer1 value to **TIMER1_CURRENTVALUE**. This register is automatically cleared after the timer is locked.	0x0

Bits	Access	Name	Description	Reset
[2]	RW	timer1_int_mask	Timer1 interrupt mask 0: not masked 1: masked	0x0
[1]	RW	timer1_mode	Timer1 mode control 0: free mode 1: periodic mode	0x0
[0]	RW	timer1_en	Timer1 enable 0: disabled 1: enabled	0x0

TIMER1_EOI

TIMER1_EOI is the timer1 interrupt clear register.

Offset Address: 0x00C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer1_eoi	Reading this register clears the timer1 interrupt.	0x0

TIMER1_INTSTATUS

TIMER1_INTSTATUS is the timer1 interrupt status register.

Offset Address: 0x010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer1_int_status	Masked interrupt status of timer1 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

TIMER2_LOADCOUNT

TIMER2_LOADCOUNT is the timer2 initial value register.

Offset Address: 0x014 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer2_loadcount	Initial value of timer2	0x00000000

TIMER2_CURRENTVALUE

TIMER2_CURRENTVALUE is the timer2 current value register.

Offset Address: 0x018 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timer2_currentvalue	Current value of timer2	0x00000000

TIMER2_CONTROLREG

TIMER2_CONTROLREG is the timer2 control register.

Offset Address: 0x01C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RW	timer2_lock	Timer2 lock control 0: not locked 1: Locks the timer2 value to **TIMER2_CURRENTVALUE**. This register is automatically cleared after the timer is locked.	0x0
[2]	RW	timer2_int_mask	Timer2 interrupt mask 0: not masked 1: masked	0x0
[1]	RW	timer2_mode	Timer2 mode control 0: free mode 1: periodic mode	0x0
[0]	RW	timer2_en	Timer2 enable 0: disabled 1: enabled	0x0

TIMER2_EOI

TIMER2_EOI is the timer2 interrupt clear register.

Offset Address: 0x020 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer2_eoi	Reading this register clears the timer2 interrupt.	0x0

TIMER2_INTSTATUS

TIMER2_INTSTATUS is the timer2 interrupt status register.

Offset Address: 0x024 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer2_int_status	Masked interrupt status of timer2 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

TIMER3_LOADCOUNT

TIMER3_LOADCOUNT is the timer3 initial value register.

Offset Address: 0x028 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer3_loadcount	Initial value of timer3	0x00000000

TIMER3_CURRENTVALUE

TIMER3_CURRENTVALUE is the timer3 current value register.

Offset Address: 0x02C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timer3_currentvalue	Current value of timer3	0x00000000

TIMER3_CONTROLREG

TIMER3_CONTROLREG is the timer3 control register.

Offset Address: 0x030 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RW	timer3_lock	Timer3 lock control 0: not locked 1: Locks the timer3 value to **TIMER3_CURRENTVALUE**. This register is automatically cleared after the timer is locked.	0x0
[2]	RW	timer3_int_mask	Timer3 interrupt mask 0: not masked 1: masked	0x0
[1]	RW	timer3_mode	Timer3 mode control 0: free mode 1: periodic mode	0x0
[0]	RW	timer3_en	Timer3 enable 0: disabled 1: enabled	0x0

TIMER3_EOI

TIMER3_EOI is the timer3 interrupt clear register.

Offset Address: 0x034 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer3_eoi	Reading this register clears the timer3 interrupt.	0x0

TIMER3_INTSTATUS

TIMER3_INTSTATUS is the timer3 interrupt status register.

Offset Address: 0x038 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer3_int_status	Masked interrupt status of timer3 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

TIMER4_LOADCOUNT

TIMER4_LOADCOUNT is the timer4 initial value register.

Offset Address: 0x03C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timer4_loadcount	Initial value of timer4	0x00000000

TIMER4_CURRENTVALUE

TIMER4_CURRENTVALUE is the timer4 current value register.

Offset Address: 0x040 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timer4_currentvalue	Current value of timer4	0x00000000

TIMER4_CONTROLREG

TIMER4_CONTROLREG is the timer4 control register.

Offset Address: 0x044 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RW	timer4_lock	Timer4 lock control 0: not locked 1: Locks the timer4 value to **TIMER4_CURRENTVALUE**. This register is automatically cleared after the timer is locked.	0x0
[2]	RW	timer4_int_mask	Timer4 interrupt mask 0: not masked 1: masked	0x0
[1]	RW	timer4_mode	Timer4 mode control 0: free mode 1: periodic mode	0x0
[0]	RW	timer4_en	Timer4 enable 0: disabled 1: enabled	0x0

TIMER4_EOI

TIMER4_EOI is the timer4 interrupt clear register.

Offset Address: 0x048 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer4_eoi	Reading this register clears the timer4 interrupt.	0x0

TIMER4_INTSTATUS

TIMER4_INTSTATUS is the timer4 interrupt status register.

Offset Address: 0x04C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer4_int_status	Masked interrupt status of timer4 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

TIMERS_INTSTATUS

TIMERS_INTSTATUS is the interrupt status register for timers.

Offset Address: 0x0A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RO	timer4_int_status_sum	Masked interrupt status of timer4 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[2]	RO	timer3_int_status_sum	Masked interrupt status of timer3 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[1]	RO	timer2_int_status_sum	Masked interrupt status of timer2 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

[0]	RO	timer1_int_status_sum	Masked interrupt status of timer1 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

TIMERS_EOI

TIMERS_EOI is the interrupt clear register for timers.

Offset Address: 0x0A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timer_seoi	Reading this register clears the timer interrupt.	0x0

TIMERS_RAWINTSTATUS

TIMERS_RAWINTSTATUS is the raw interrupt status register for timers.

Offset Address: 0x0A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RO	timer4_raw_int_status	Status of the unmasked interrupt of timer4 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[2]	RO	timer3_raw_int_status	Unmasked interrupt status for timer3 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[1]	RO	timer2_raw_int_status	Status of the unmasked interrupt of timer2 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[0]	RO	timer1_raw_int_status	Status of the unmasked interrupt of timer1 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

2.9 Timer

2.9.1 Overview

The chip has four identical timers that can be enabled and configured separately. The clock intervals can also be flexibly specified. A timer counts down from the preset value to 0, triggering an interrupt. The initial data, working mode, and interrupt mode of each timer can be configured by using registers.

2.9.2 Features

The timer has the following features:

- Four 32-bit timer units that can be separately configured are provided.
- Free mode and periodic mode are supported.
- Each timer unit can be enabled separately.

2.9.3 Operating Modes

The working clock frequency of each timer can be 40 MHz or 24 MHz. Frequency division is supported

A timer runs in free mode or periodic mode. Each timer can be configured separately.

- In free mode, a timer counts from 0xFFFF_FFFF to 0 in descending order.
- In periodic mode, a timer counts down from the value of **TIMER*x*_LOADCOUNT** (*x* = 0–3) to **0**.

The following uses timer1 and timer2 as examples to describe the timer workflow.

Step 1 Configure TIMER1_CONTROLREG and TIMER2_CONTROLREG to initialize the timers.

- Write **0** to TIMER1_CONTROLREG[timer1_en] and TIMER2_CONTROLREG[timer2_en] to disable the timers.

  Note: To avoid absence of synchronization, the timers must be disabled before setting TIMER1_LOADCOUNT and TIMER2_LOADCOUNT.
- Write **0** to TIMER1_CONTROLREG[timer1_mode] and TIMER2_CONTROLREG[timer2_mode] to set the timers to the free mode. Or write **1** to set the timers to periodic mode.

  Note: Before setting the RTC working mode to free mode, you must set all valid bits of TIMER1_LOADCOUNT and TIMER2_LOADCOUNT to 1.
- Write **1** to TIMER1_CONTROLREG[timer1_int_mask] and TIMER2_CONTROLREG[timer2_int_mask] to enable the interrupt mask. Or write **0** to disable the interrupt mask.

Step 2 Write the initial values of the timers to TIMER1_LOADCOUNT and TIMER2_LOADCOUNT.

Step 3 Write **1** to TIMER1_CONTROLREG[timer1_en] and TIMER2_CONTROLREG[timer2_en] to enable the timers and start the countdown.

----End

2.9.4 Register Summary

Table 2-6 describes the timer registers.

Table 2-6 Summary of timer registers (base address: 0x4005_0000)

Offset Address	Register	Description	Page
0x000 + 0x14 * *n*	TIMERn_LOADCOUNT	Initial value register for timer *n*	2-22
0x004 + 0x14 * *n*	TIMERn_CURRENTVALUE	Current value register for timer *n*	2-23
0x008 + 0x14 * *n*	TIMERn_CONTROLREG	Control register for Timer n	2-23
0x00C + 0x14 * *n*	TIMERn_EOI	Interrupt clear register for timer *n*	2-23
0x010 + 0x14 * *n*	TIMERn_INTSTATUS	Interrupt status register for timer *n*	2-24
0x0A0	TIMERS_INTSTATUS	Interrupt status register for timers	2-19
0x0A4	TIMERS_EOI	Interrupt clear register for timers	2-20
0x0A8	TIMERS_RAWINTSTATUS	Raw interrupt status register	2-20

Table 2-7 describes the value ranges and meanings of the variable in the offset addresses for timer registers.

Table 2-7 Variable in the offset addresses for timer registers

Variable	Value Range	Description
n	0–3	Timer ID

2.9.5 Register Description

TIMERn_LOADCOUNT

TIMERn_LOADCOUNT is the initial value register for timer *n*.

Offset Address: 0x000 + 0x14 * *n* Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	timern_loadcount	Initial value of timer *n*	0x00000000

TIMERn_CURRENTVALUE

TIMERn_CURRENTVALUE is the current value register for timer *n*.

Offset Address: 0x004 + 0x14 * *n* Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RO	timern_currentvalue	Current value of timer *n*	0x00000000

TIMERn_CONTROLREG

TIMERn_CONTROLREG is the control register for timer *n*.

Offset Address: 0x008 + 0x14 * *n* Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RW	timern_lock	Lock control for timer *n* 0: not locked 1: Locks the current value of timer *n* to **TIMER*n*_CURRENTVALUE**. **Note: This register is automatically cleared after the timer is locked.**	0x0
[2]	RW	timern_int_mask	Timer *n* interrupt mask 0: not masked 1: masked	0x0
[1]	RW	timern_mode	Timer *n* mode 0: free mode 1: periodic mode	0x0
[0]	RW	timern_en	Timer *n* enable 0: disabled 1: enabled	0x0

TIMERn_EOI

TIMERn_EOI is the interrupt clear register for timer *n*.

Offset Address: 0x00C + 0x14 * *n* Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timern_eoi	Reading this register clears a Timer n	0x0

			interrupt.	

TIMERn_INTSTATUS

TIMERn_INTSTATUS is the interrupt status register for timer *n*.

Offset Address: 0x010 + 0x14 * *n* Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timern_int_status	Masked interrupt status of timer *n* 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

TIMERS_INTSTATUS

TIMERS_INTSTATUS is the interrupt status register for timers.

Offset Address: 0x0A0 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RO	timer3_int_status_sum	Masked interrupt status of timer3 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[2]	RO	timer2_int_status_sum	Masked interrupt status of timer2 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[1]	RO	timer1_int_status_sum	Masked interrupt status of timer1 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[0]	RO	timer0_int_status_sum	Masked interrupt status of timer0 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

TIMERS_EOI

TIMERS_EOI is the interrupt clear register for timers.

Offset Address: 0x0A4 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	timers_eoi_sum	Reading this register clears all timer interrupts.	0x0

TIMERS_RAWINTSTATUS

TIMERS_RAWINTSTATUS is the raw interrupt status register for timers.

Offset Address: 0x0A8 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RO	timer3_raw_int_status	Unmasked interrupt status for timer3 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[2]	RO	timer2_raw_int_status	Unmasked interrupt status for timer2 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[1]	RO	timer1_raw_int_status	Unmasked interrupt status for timer1 0: The interrupt is invalid. 1: The interrupt is valid.	0x0
[0]	RO	timer0_raw_int_status	Unmasked interrupt status for timer0 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

2.10 WDT

2.10.1 Overview

During normal operation, the system resets the watchdog timer (WDT) at a specified interval (which is programmable) to prevent it from timing out. If the system fails to reset the WDT due to a hardware fault or program error, the timer will generate a system reset signal.

2.10.2 Features

A 32-bit built-in and programmable WDT is provided, which has the following features:

- The WDT times out when it counts down to 0.
- The WDT runs in either of the following two modes:

  - Mode 1: A system reset is generated upon each WDT timeout.
  - Mode 2: When the first timeout occurs, the WDT generates an interrupt. If the interrupt is not cleared until the second timeout, a system reset is generated.
- The timeout period can be configured.

2.10.3 Operating Modes

The reset value of the WDT is **0xFFFF**, which indicates that the WDT counts down from **0xFFFF** initially. When the WDT is reset or the WDT runs in mode 2, the WDT starts to count down from the configured value (timeout period) after the WDT counts down to **0** from **0xFFFF** for the first time. If the WDT needs to count down from the configured value, it must be reset immediately after being enabled.

After the WDT is started, writing **0** to SOFT_RESET[soft_rst_wdt_n] or asserting a global reset can pause the WDT.

To start the WDT, perform the following steps:

Step 1 Write **1** to CLKEN[wdt_clken] to enable the WDT.

Step 2 Write the WDT timeout period to WDT_TORR (in clock cycles).

Step 3 Write WDT_CR[rmod] to set the WDT working mode.

Step 4 Write **1** to WDT_CR[wdt_en] to start the WDT.

----End

2.10.4 Register Summary

Table 2-8 describes the WDT registers.

Table 2-8 Summary of WDT registers (base address: 0x4000_0000)

Offset Address	Register	Description	Page
0x000	WDT_CR	WDT control register	2-26
0x004	WDT_TORR	Timeout period register	2-27
0x008	WDT_CCVR	Current WDT value register	2-27
0x00C	WDT_CRR	Counter restart register	2-28
0x010	WDT_STAT	Interrupt state register	2-28
0x014	WDT_EOI	Interrupt clear register	2-28

2.10.5 Register Description

WDT_CR

WDT_CR is the WDT control register.

Offset Address: 0x000 Total Reset Value: 0x0000_0008

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x0000000
[5]	RW	cr_bit5	Meaningless	0x0
[4:2]	RW	reserved	Reserved	0x2
[1]	RW	rmod	Reset mode select 0: A system reset is generated upon each WDT timeout. 1: When the first timeout occurs, the WDT generates an interrupt. If the interrupt is not cleared until the second timeout, a system reset is generated.	0x0
[0]	RWS	wdt_en	WDT enable 0: disabled 1: enabled **Note: Once this bit is set to 1, the WDT can be cleared only by system reset or soft reset.**	0x0

WDT_TORR

WDT_TORR is the timeout period register.

Offset Address: 0x004 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	RW	top	WDT interval (in clock cycles) Calculation formula: Timeout period = 0xFFFF x 2^top = 2^(top + 16) **Note: The WDT counts down from 0xFFFF but not the configured value after the WDT is reset. To have the WDT count from the configured value, the WDT must be reset.**	0x0

WDT_CCVR

WDT_CCVR is the current WDT value register.

Offset Address: 0x008 Total Reset Value: 0x0000_FFFF

Bits	Access	Name	Description	Reset
[31:0]	RO	ccvr	Current value of the WDT	0x0000FFFF

WDT_CRR

WDT_CRR is the WDT restart register.

Offset Address: 0x00C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	WO	crr	Writing **0x76** to this register restarts the WDT.	0x00

WDT_STAT

WDT_STAT is the interrupt status register.

Offset Address: 0x010 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	isr	Masked interrupt status 0: The interrupt is invalid. 1: The interrupt is valid.	0x0

WDT_EOI

WDT_EOI is the interrupt clear register.

Offset Address: 0x014 Total Reset Value: 0x0000_0000

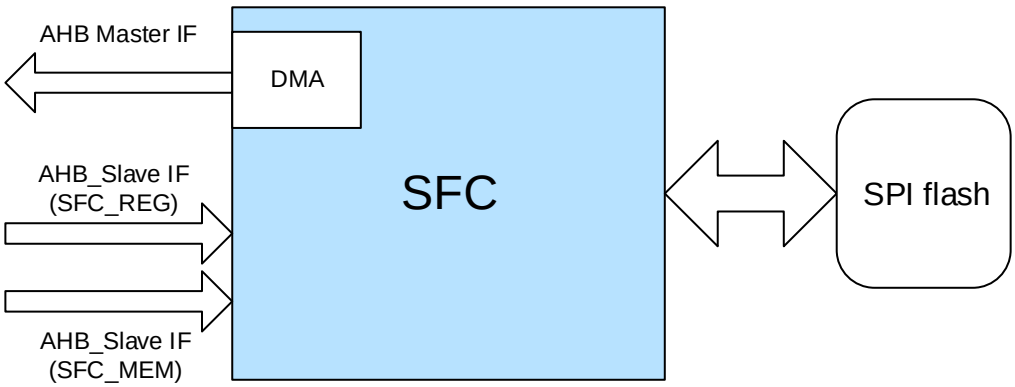
Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	eoi	Interrupt clear. The interrupt is cleared by reading this register.	0x0

3 SFC

3.1 Overview

The serial peripheral interface flash controller (SFC) provides an advanced high-performance bus (AHB) slave interface to the service side to facilitate AHB's control on access to the SPI flash and provides an AHB master interface to read and write to the SPI flash in direct memory access (DMA) mode.

Figure 3-1 SFC application block diagram



Note: IF is short for the interface.

3.2 Features

AHB Slave Interface

The AHB slave interface has the following features:

- Access to internal configuration registers through selection signals or direct access to the SPI flash memory
- AMBA 2.0 specifications
- Little endian mode only

AHB Master Interface

The AHB master interface has the following features:

- Data transfer between the memory and SPI flash in DMA mode
- AMBA 2.0 specifications
- Little endian mode only
- Four transfer types: Single, INCR4, INCR8, and INCR16
- Early termination not supported
- Bus lock

Memory Interfaces

The memory interfaces have the following features:

- CS: A maximum of 16-Mbit (3-byte address mode) storage space is supported. A CS can be mapped to the system address space with a configurable base address. Only CS 1 is supported. CS 0 is not supported.
- Address aliasing: Address 0 can be mapped to the SPI flash address aliasing after power-on to boot the chip from the SPI flash. Address aliasing is fixed at CS 1.
- Five interface types: standard SPI, dual-O/dual-I SPI, quad-O/quad-I SPI, dual-I/O SPI, and quad-I/O SPI. The standard SPI is used by default after power-on. The interface type can be changed by configuring the register.
- Execute in place (XIP) not supported
- SPI flash reading and writing by means of the bus, register programming, and DMA
- Various write-protect operations
- SPI mode 0 and mode 3: According to the protocol specifications, the SPI flash memory that supports SPI mode 0 and mode 3 samples data on the rising edge of the clock and outputs data on the falling edge of the clock.

Flash Data Scrambling/Descrambling

Flash data scrambling/descrambling has the following features:

- Scrambles data written to the flash memory and descrambles data read from the flash memory.
- Data read from a specified address range of the flash memory can be separately not descrambled.

NOTE

The eFUSE controls whether to enable flash data scrambling/descrambling by using the SFC as well as the scrambling/descrambling parameters. To ensure that the flash data is read and written consistently, during the flash read/write process, the scrambling/descrambling enable state and parameter settings output by the eFUSE must remain unchanged.

3.3 Operating Modes

3.3.1 Reading and Writing the Flash Memory

The SPI flash memory can be read and written in the following ways:

- Read/Write the SPI flash memory by sending instructions such as SPI flash program and read based on register configurations. For example, write **0x0000_7F8B** to CMD_CONFIG to read 64-byte data.

  In this mode, the flash instructions to be sent are directly controlled.
- Read/Write the SPI flash memory over the AHB slave interface.

  The SFC automatically maps the read/write timing on the AHB bus to an SPI flash read/write instruction.
- Read/Write the SPI flash memory by copying data between the flash memory and external memory in DMA mode.

3.3.2 Other Operations

Other operations such as erasing the flash memory, taking the flash memory to the deep power-down mode, and reading the device ID can be implemented by writing an instruction to CMD_INS[REG_INS]. For details, see the flash memory data sheet.

For example, write **0x0000_009F** to CMD_INS to read the device ID.

3.3.3 Initialization Workflow

NOTICE

- The operation registers do not need to be initialized but need to be reconfigured for each operation.
- Note: The following initialization workflow is for reference only.

The initialization workflow is as follows:

Step 1 Configure the TIMING register if the timing parameter needs to be adjusted.

Step 2 Configure the bus operation mode registers.

- Configure BUS_FLASH_SIZE[flash_size_cs0] based on the flash memory size, which can be queried by using the **Read ID** instruction.
- Configure BUS_BASE_ADDR_CS0, BUS_BASE_ADDR_CS1, and BUS_ALIAS_CS based on the mapping between the system address space and the flash memory. The mapped space must be within the address space allocated to SFC_MEM by the system bus.

  Typically, BUS_ALIAS_CS reflects the mapped address when the system boots from the flash memory. Only the default value is valid. Therefore, the address is fixed.
- Write a configuration instruction to CMD_INS as required. (For some SPI flash memories, special configurations are required to use non-standard SPI read and write timings.)
- Configure the read and write instructions and parameters for the bus through BUS_CONFIG1 or BUS_CONFIG2.

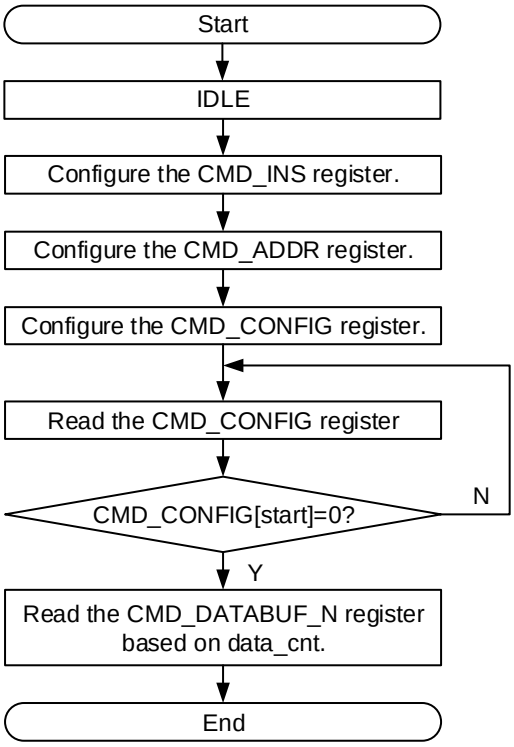
  For example, write **0xCC85_EB1E** to BUS_CONFIG1 to configure write instruction 32 in write mode quad-input SPI and read instruction EB in read mode quad-I/O SPI.
- Write **1** to BUS_CONFIG1[wr_enable] if the bus write operation is required. The bus write function is disabled by default.

----End

3.3.4 Reading the SPI Flash by Using Registers

Figure 3-2 shows the process of reading the flash memory by using registers in polled mode.

Figure 3-2 Reading the SPI flash by using registers in polled mode



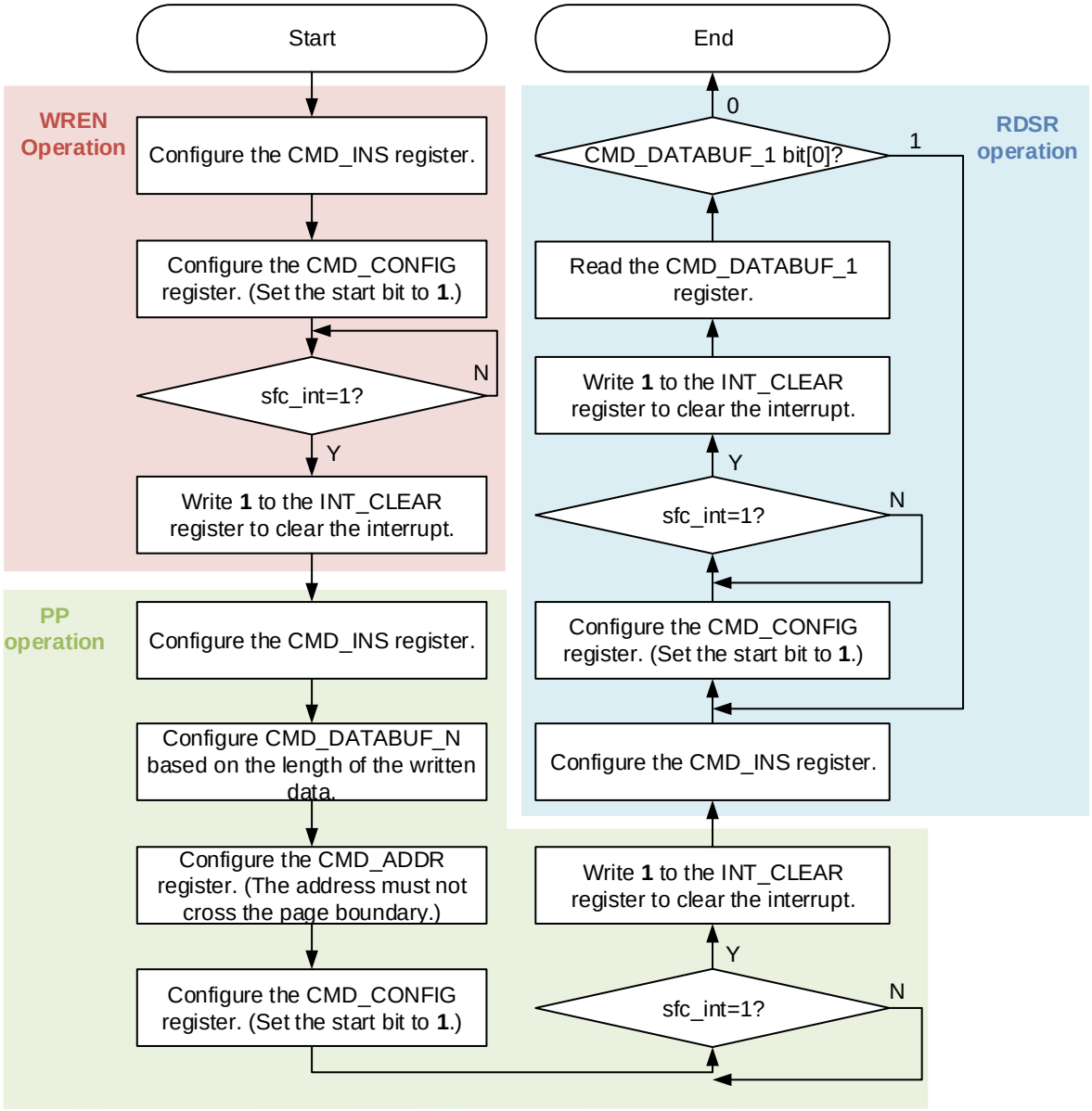
3.3.5 Writing to the SPI Flash by Using Registers

NOTICE

- When the flash data is being written by using registers, access to the flash memory by means of bus and DMA is not allowed.
- Writing across the page boundaries is not allowed. (Writing to the SPI flash by using registers is not cross-page-boundary protected, which should be guaranteed by software. If the 256-byte boundary is crossed, the data will wrap around to the beginning of the page, overwriting the data previously stored.)

Figure 3-3 shows the process of writing to the flash memory by using registers in interrupt-driven mode.

Figure 3-3 Writing to the SPI flash by using registers in interrupt-driven mode

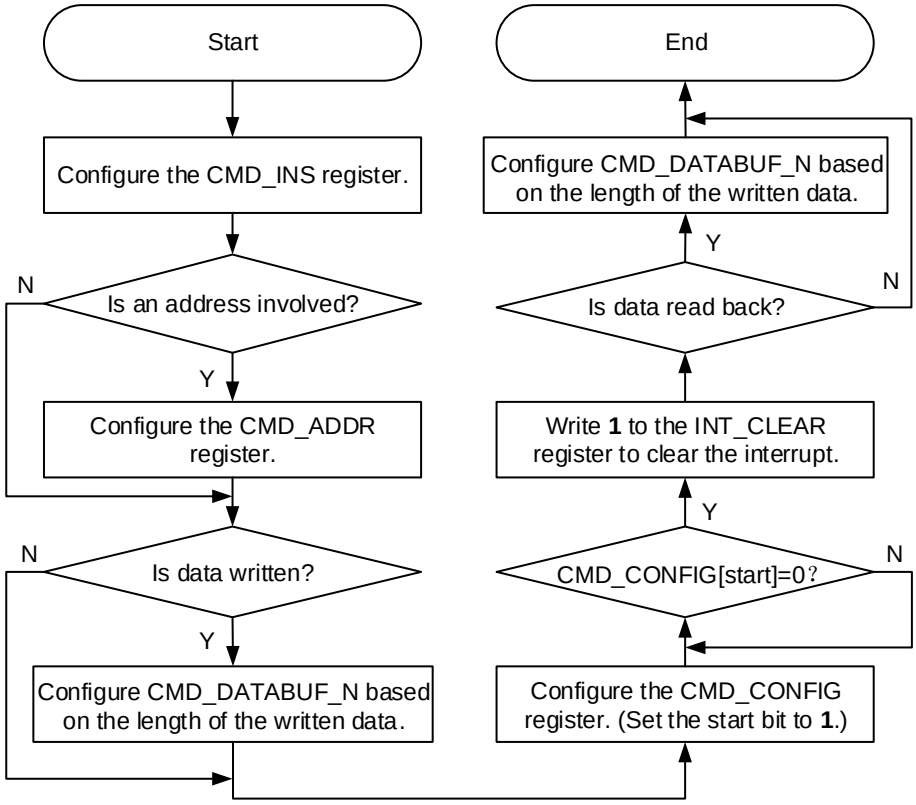


Note: WREN is short for Write Read Enable, PP is short for Page Program, and RDSR is short for the Read Status Register.

3.3.6 Performing Other Operations on the SPI Flash by Using Registers

Figure 3-4 shows the process of performing other operations by using registers.

Figure 3-4 Performing other operations on the SPI flash by using registers



NOTE

The SFC does not support the combined SPI timing (1 opcode byte + 3 all-zero dummy bytes). If some flash instructions require such combined timing, use the timing (1 opcode byte + 3 all-zero ADDR bytes) for replacement.

3.3.7 Reading/Writing to the SPI Flash over the AHB Slave Interface

After SFC POR, it is reset to the standard SPI timing mode. In this case, the SPI flash memory can be directly read without additional configurations.

By default, writing to the SPI flash over the AHB slave interface is not allowed. To enable this function, write **1** to BUS_CONFIG1[wr_enable].

For details about how to modify the default settings, see 3.3.3 Initialization Workflow.

3.3.8 Reading/Writing to the SPI Flash in DMA Mode

To read and write to the SPI flash in DMA mode, perform the following steps:

Step 1 For details about how to adjust the bus operation timing, see section 3.3.3 Initialization Workflow.

Step 2 Write the memory start address for the DMA operation to BUS_DMA_MEM_SADDR. Write the flash start address (flash offset address) to BUS_DMA_FLASH_SADDR. Write the data length to BUS_DMA_LEN.

Step 3 Write BUS_DMA_CTRL to configure the read/write direction and CS (flash 0 or flash 1).

Step 4 Write **1** to BUS_DMA_CTRL[start] to enable the DMA operation.

Step 5 Wait until the dma_done interrupt is triggered (interrupt-driven mode) or polled DMA polling is complete (that is, BUS_DMA_CTRL[start] changes to **0**).

NOTE

- The SPI flash can be accessed by issuing instructions through registers during DMA operations.
- The SPI flash can be directly accessed over the AHB slave interface during DMA operations. However, the bus operation settings must remain unchanged during the process.
- Ensure that the start address is 4-byte aligned during DMA operations.

----End

3.3.9 Reading Data after Erasing the Flash Memory

If SFC scrambling/descrambling is enabled, after the software erases the flash memory, the software needs to read the data of the erased address for verification. If **0xFF** is read, the erase operation is successful (the read data must not be descrambled). The process is as follows:

Step 1 Write SFC_NOSCM_ADDR_START[sfc_noscm_addr_start] and SFC_NOSCM_ADDR_END[sfc_noscm_addr_end] to configure the start address and end address for the software to read the flash memory.

Note: The start address and end address are relative addresses of the flash memory.

Step 2 Write **1** to SFC_NOSCM_EN[sfc_noscm_en] to disable descrambling of the data read from the address range configured in Step 1.

Step 3 The software reads the flash data for judgment.

----End

3.4 Register Summary

Table 3-1 lists the SFC registers.

Table 3-1 Summary of SFC registers (base address: 0x4080_0000)

Offset Address	Register	Description	Page
0x0100	GLOBAL_CONFIG	Global configuration register	3-9
0x0110	TIMING	Timing configuration register	3-10
0x0120	INT_RAW_STATUS	Raw interrupt status register	3-10
0x0124	INT_STATUS	Masked interrupt status register	3-11
0x0128	INT_MASK	Interrupt mask register	3-11
0x012C	INT_CLEAR	Interrupt clear register	3-11

Offset Address	Register	Description	Page
0x0150	SFC_NOSCM_ADDR_START	Start address register for read data non-descrambling	3-12
0x0154	SFC_NOSCM_ADDR_END	End address register for read data non-descrambling	3-12
0x0158	SFC_NOSCM_EN	Read data non-descrambling enable register	3-13
0x01F8	VERSION	Version register	3-13
0x01FC	VERSION_SEL	Version selection register	3-13
0x0200	BUS_CONFIG1	Bus operation configuration register 1	3-14
0x0204	BUS_CONFIG2	Bus operation configuration register 2	3-15
0x0210	BUS_FLASH_SIZE	Mapping size register for bus operations	3-16
0x0214	BUS_BASE_ADDR_CS0	CS 0 mapped base address register for bus operations	3-17
0x0218	BUS_BASE_ADDR_CS1	CS 1 mapped base address register for bus operations	3-17
0x021C	BUS_ALIAS_ADDR	Alias mapped base address register for bus operations	3-17
0x0220	BUS_ALIAS_CS	Alias CS indication register for bus operations	3-17
0x0240	BUS_DMA_CTRL	DMA operation control register	3-18
0x0244	BUS_DMA_MEM_SADDR	DDR start address register for DMA operations	3-18
0x0248	BUS_DMA_FLASH_SADDR	SPI flash start address register for DMA operations	3-19
0x024C	BUS_DMA_LEN	Data transfer length register for DMA operations	3-19
0x0250	BUS_DMA_AHB_CTRL	Burst mode selection register for DMA operations on AHB	3-19
0x0300	CMD_CONFIG	Instruction operation configuration register	3-20
0x0308	CMD_INS	Instruction register for instruction operations through registers	3-21
0x030C	CMD_ADDR	Address register for instruction operations through registers	3-21
0x0400 + 4 * *n*	CMD_DATABUF_N	Data buffer register for instruction operations through registers	3-21

Table 3-2 describes the value range and meaning of the variable in the offset addresses of SFC registers.

Table 3-2 Variable in the offset addresses of SFC registers

Variable	Value Range	Description
n	0–15	Data buffer

3.5 Register Description

GLOBAL_CONFIG

GLOBAL_CONFIG is the global configuration register.

Offset Address: 0x0100 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x0000000
[5:3]	RW	rd_delay	Delay for reading data from the SPI flash memory 000: 0.5–1 clock cycles (default) 001: 1–1.5 clock cycles 010: 1.5–2 clock cycles 011: 2–2.5 clock cycles 100: 2.5–3 clock cycles 101: 3–3.5 clock cycles 110: 3.5–4 clock cycles 111: not supported (equivalent to value **110**)	0x0
[2]	RW	flash_addr_mode	SPI addressing mode 0: 3-byte addressing mode (default) 1: 4-byte addressing mode **Note: When CMD_CONFIG[start] is 1, writing to this bit has no effect.**	0x0
[1]	RW	wp_en	Hardware-based write-protect enable (by using the write-protect pin) 0: disabled 1: enabled	0x0
[0]	RW	mode	SPI mode configuration 0: mode 0 1: mode 3	0x0

TIMING

TIMING is the timing configuration register.

Offset Address: 0x0110 Total Reset Value: 0x0000_660F

Bits	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:12]	RW	tcsh	CS hold time 0x0–0x7: (*n* + 1) clock cycles For example, **0x6** indicates 7 clock cycles.	0x6
[11]	-	reserved	Reserved	0x0
[10:8]	RW	tcss	CS setup time 0x0–0x7: (*n* + 1) clock cycles For example, **0x6** indicates 7 clock cycles.	0x6
[7:4]	-	reserved	Reserved	0x0
[3:0]	RW	tshsl	Sets the interval between two flash operations. 0x0–0xF: (*n* +2) clock cycles For example, **0xF** indicates 17 clock cycles.	0xF

INT_RAW_STATUS

INT_RAW_STATUS is the raw interrupt status register.

Offset Address: 0x0120 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RO	dma_done_int_raw_status	Raw interrupt status for DMA operation completion (prior to masking) 0: uncompleted 1: completed	0x0
[0]	RO	cmd_op_end_raw_status	Raw interrupt status for instruction operation completion (prior to masking) 0: uncompleted 1: completed	0x0

INT_STATUS

INT_STATUS is the masked interrupt status register.

Offset Address: 0x0124 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RO	dma_done_int_status	Masked interrupt status for DMA operation completion 0: uncompleted 1: completed	0x0
[0]	RO	cmd_op_end_status	Masked interrupt status for instruction operation completion 0: uncompleted 1: completed	0x0

INT_MASK

INT_MASK is the interrupt mask register.

Offset Address: 0x0128 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RW	dma_done_int_mask	Interrupt mask for DMA operation completion 0: masked 1: not masked	0x0
[0]	RW	cmd_op_end_int_mask	Interrupt mask for instruction operation completion 0: masked 1: not masked	0x0

INT_CLEAR

INT_CLEAR is the interrupt clear register.

Offset Address: 0x012C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	WO	dma_done_int_clr	Interrupt clear for DMA operation completion. Writing **1** clears	0x0

Bits	Access	Name	Description	Reset
			INT_STATUS[dma_done_int_status] and INT_RAW_STATUS[dma_done_int_raw_status]. 0: not cleared 1: cleared **Note: This bit is automatically set back to 0 after the clear operation is complete.**	
[0]	WO	cmd_op_end_int_clr	Interrupt clear for instruction operation completion. Writing **1** clears INT_STATUS[cmd_op_end_status] and INT_RAW_STATUS[cmd_op_end_raw_status]. 0: not cleared 1: cleared **Note: This bit is automatically set back to 0 after the clear operation is complete.**	0x0

SFC_NOSCM_ADDR_START

SFC_NOSCM_ADDR_START is the start address register for read data non-descrambling.

Offset Address: 0x0150 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	sfc_noscm_addr_start	Start address of the read data that is not descrambled After the software erases the flash memory, the software reads back data that is not descrambled for verification. The data start and end addresses need to be configured for read data non-descrambling.	0x000000

SFC_NOSCM_ADDR_END

SFC_NOSCM_ADDR_END is the end address register for read data non-descrambling.

Offset Address: 0x0154 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:24]	-	reserved	Reserved	0x00
[23:0]	RW	sfc_noscm_addr_end	End address of the read data that is not descrambled	0x000000

Bits	Access	Name	Description	Reset
			After the software erases the flash memory, the software reads back data that is not descrambled for verification. The data start and end addresses need to be configured for read data non-descrambling.	

SFC_NOSCM_EN

SFC_NOSCM_EN is the read data non-descrambling enable register.

Offset Address: 0x0158 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	sfc_noscm_en	Read data non-descrambling enable 0: disabled 1: enabled After the software erases the flash memory, the software reads back data that is not descrambled for verification. The data start and end addresses need to be configured for read data non-descrambling.	0x0

VERSION

VERSION is the version register.

Offset Address: 0x01F8 Total Reset Value: 0x0000_0350

Bits	Access	Name	Description	Reset
[31:0]	RO	version	SFC version: V350	0x00000350

VERSION_SEL

VERSION_SEL is the version selection register.

Offset Address: 0x01FC Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	version_sel	Register group version indicator 0: register group of an earlier version	0x1

Bits	Access	Name	Description	Reset
			1: register group of a later version **Note: The value is fixed to 1 and cannot be changed.**	

BUS_CONFIG1

BUS_CONFIG1 is bus operation mode configuration register 1.

Offset Address: 0x0200 Total Reset Value: 0x8080_0300

Bits	Access	Name	Description	Reset
[31]	RW	rd_enable	Bus read enable 0: disabled 1: enabled	0x1
[30]	RW	wr_enable	Bus write enable 0: disabled 1: enabled	0x0
[29:22]	RW	wr_ins	Write instruction	0x02
[21:19]	RW	wr_dummy_bytes	Dummy byte count of the bus write operation 000: 0 001: 1 010: 2 ... 111: 7	0x0
[18:16]	RW	wr_mem_if_type	Type of the SPI flash interface for the bus write operation 000: standard SPI 001: dual-I/dual-O SPI 010: dual-I/O SPI 011: full dual-I/O SPI 100: reserved 101: quad-I/dual-O SPI 110: quad-I/O SPI 111: full quad-I/O SPI	0x0
[15:8]	RW	rd_ins	Read instruction	0x03
[7:6]	RW	rd_prefetch_cnt	Clock cycles for prefetching data when the flash memory is read at a variable data length through the bus 00: prefetch disabled (default)	0x0

Bits	Access	Name	Description	Reset
			01: 1 clock cycle 10: 2 clock cycles 11: 3 clock cycles	
[5:3]	RW	rd_dummy_bytes	Dummy byte count of the bus read operation 00: 0 001: 1 010: 2 ... 111: 7	0x0
[2:0]	RW	rd_mem_if_type	Type of the SPI flash interface for the bus read operation 000: standard SPI 001: dual-I/dual-O SPI 010: dual-I/O SPI 101: quad-I/dual-O SPI 110: quad-I/O SPI Other values: reserved	0x0

BUS_CONFIG2

BUS_CONFIG2 is bus operation mode configuration register 2.

Offset Address: 0x0204 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	wip_locate	Location of the write in progress (WIP) bit in the flash status register 000: bit[0] (default) 001: bit[1] 010: bit[2] 011: bit[3] 100: bit[4] 101: bit[5] 110: bit[6] 111: bit[7]	0x0

BUS_FLASH_SIZE

BUS_FLASH_SIZE is the mapping size register for bus operations.

Offset Address: 0x0210 Total Reset Value: 0x0000_0909

Bits	Access	Name	Description	Reset
[31:12]	-	reserved	Reserved	0x00000
[11:8]	RW	flash_size_cs1	Capacity of the SPI flash memory connected to CS 1 0x0: no SPI flash memory connected 0x1: 512 kbits 0x2: 1 Mbit 0x3: 2 Mbits 0x4: 4 Mbits 0x5: 8 Mbits 0x6: 16 Mbits 0x7: 32 Mbits 0x8: 64 Mbits 0x9: 128 Mbits (default) 0xA: 256 Mbits 0xB: 512 Mbits 0xC: 1 Gbit 0xD: 2 Gbits 0xE: 4 Gbits 0xF: 8 Gbits	0x9
[7:4]	-	reserved	Reserved	0x0
[3:0]	RW	flash_size_cs0	Capacity of the SPI flash memory connected to CS 0 0x0: no SPI flash memory connected 0x1: 512 kbits 0x2: 1 Mbit 0x3: 2 Mbits 0x4: 4 Mbits 0x5: 8 Mbits 0x6: 16 Mbits 0x7: 32 Mbits 0x8: 64 Mbits 0x9: 128 Mbits (default) 0xA: 256 Mbits 0xB: 512 Mbits 0xC: 1 Gbit	0x9

Bits	Access	Name	Description	Reset
			0xD: 2 Gbits 0xE: 4 Gbits 0xF: 8 Gbits	

BUS_BASE_ADDR_CS0

BUS_BASE_ADDR_CS0 is the CS 0 mapped base address register for bus operations.

Offset Address: 0x0214 Total Reset Value: 0x3200_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	bus_base_addr_high_cs0	Base address of the system space to which the CS 0 flash is mapped	0x3200
[15:0]	-	reserved	Reserved	0x0000

BUS_BASE_ADDR_CS1

BUS_BASE_ADDR_CS1 is the CS 1 mapped base address register for bus operations.

Offset Address: 0x0218 Total Reset Value: 0x0040_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	bus_base_addr_high_cs1	Base address of the system space to which the CS 1 flash is mapped	0x0040
[15:0]	-	reserved	Reserved	0x0000

BUS_ALIAS_ADDR

BUS_ALIAS_ADDR is the alias mapped base address register for bus operations.

Offset Address: 0x021C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:16]	RW	flash_alias_addr	Alias base address of the system space to which the flash is mapped	0x0000
[15:0]	-	reserved	Reserved	0x0000

BUS_ALIAS_CS

BUS_ALIAS_CS is the alias CS indication register for bus operations.

Offset Address: 0x0220 Total Reset Value: 0x0000_0001

Bits	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RO	alias_cs	Alias CS indicator, controlled by an external pin and retained after reset 0: CS 0 1: CS 1	0x1

BUS_DMA_CTRL

BUS_DMA_CTRL is the DMA operation control register.

Offset Address: 0x0240 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x0000000
[4]	RW	dma_sel_cs	CS specified for the DMA operation 0: CS 0 1: CS 1	0x0
[3:2]	-	reserved	Reserved	0x0
[1]	RW	dma_rw	DMA read/write indicator 0: write operation 1: read operation	0x0
[0]	RW	dma_start	DMA transfer enable 0: disabled 1: enabled **Note: This bit is automatically set back to 0 after the DMA transfer is complete.**	0x0

BUS_DMA_MEM_SADDR

BUS_DMA_MEM_SADDR is the DDR start address register for DMA operations.

Offset Address: 0x0244 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_mem_saddr	Memory start address for a DMA operation Value range: 0x0200_0000–0x0203_FFFF.	0x00000000

BUS_DMA_FLASH_SADDR

BUS_DMA_FLASH_SADDR is the SPI flash start address register for DMA operations.

Offset Address: 0x0248 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	dma_flash_saddr	Start address for the DMA operation on the flash memory	0x00000000

BUS_DMA_LEN

BUS_DMA_LEN is the data transfer length register for DMA operations.

Offset Address: 0x024C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	Reserved	0x0
[29:0]	RW	dma_len	Length of the data transferred during the DMA operation (*n* + 1), in bytes For example, the value **6** indicates that the length is 7 bytes.	0x00000000

BUS_DMA_AHB_CTRL

BUS_DMA_AHB_CTRL is the burst mode selection register for DMA operations on AHB.

Offset Address: 0x0250 Total Reset Value: 0x0000_0007

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2]	RW	incr16_en	INC16 burst type enable 0: disabled 1: enabled	0x1
[1]	RW	incr8_en	INC8 burst type enable 0: disabled 1: enabled	0x1
[0]	RW	incr4_en	INC4 burst type enable 0: disabled 1: enabled	0x1

CMD_CONFIG

CMD_CONFIG is the instruction operation configuration register.

Offset Address: 0x0300 Total Reset Value: 0x0000_7E00

Bits	Access	Name	Description	Reset
[31:20]	-	reserved	Reserved	0x000
[19:17]	RW	mem_if_type	Type of the SPI flash interface for the instruction operation 000: standard SPI 001: dual-I/dual-O SPI 010: dual-I/O SPI 101: quad-I/dual-O SPI 110: quad-I/O SPI Other values: reserved	0x0
[16:15]	-	reserved	Reserved	0x0
[14:9]	RW	data_cnt	Length of the read/write data, in bytes 0x00–0x3F: (*n* + 1) bytes For example, the value **0x3F** indicates 64 bytes.	0x3F
[8]	RW	rw	Read/Write indicator, with data_en set to **1** 0: write, with data TX 1: read, with data RX	0x0
[7]	RW	data_en	Whether data is involved in the operation 0: no 1: yes	0x0
[6:4]	RW	dummy_byte_cnt	Dummy byte count in the instruction operation 000: 0 001: 1 010: 2 ... 111: 7	0x0
[3]	RW	addr_en	Whether an address is involved in the operation 0: no 1: yes	0x0
[2]	-	reserved	Reserved	0x0
[1]	RW	sel_cs	CS 0: CS 0 1: CS 1	0x0

Bits	Access	Name	Description	Reset
[0]	RW	start	Instruction operation start indicator 0: end 1: start **Note: This bit is automatically set back to 0 after this operation is complete.**	0x0

CMD_INS

CMD_INS is the instruction register for instruction operations through registers.

Offset Address: 0x0308 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	reg_ins	Instruction code for the instruction operation	0x00

CMD_ADDR

CMD_ADDR is the address register for instruction operations through registers.

Offset Address: 0x030C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:30]	-	reserved	Reserved	0x0
[29:0]	RW	cmd_addr	Operation address for the instruction operation	0x00000000

CMD_DATABUF_N

CMD_DATABUF_N is the buffer register for instruction operations through registers.

Offset Address: 0x0400 + 4 * *n* Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:0]	RW	cmd_databuf_n	Data buffer *n* (*n* ranges from 0 to 15) for the instruction operation	0x00000000

4 Wi-Fi

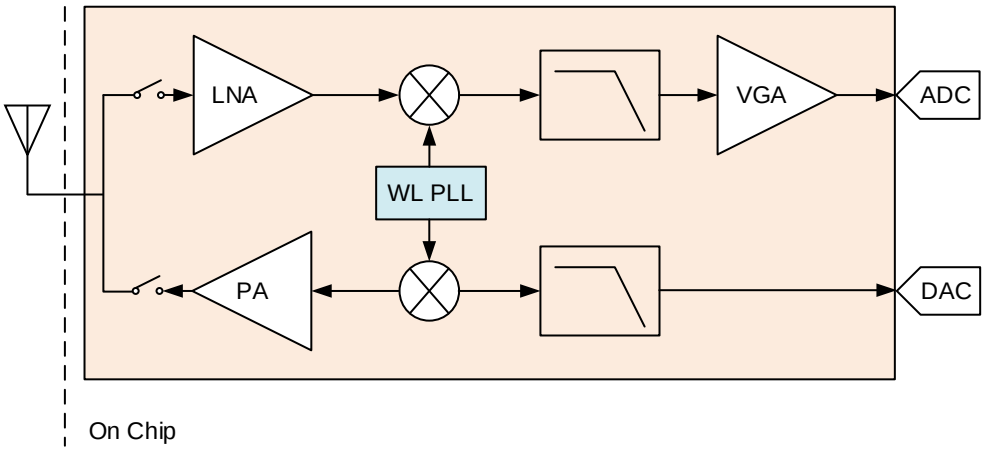
4.1 Wi-Fi RF

4.1.1 Overview

The RF IP of Hi3861 V100, Hi3861L V100, and Hi3881 V100 consists of the 2.4 GHz RX, TX, and PLL modules and supports IEEE 802.11b/g/n (20 MHz). The RF circuit has the following functions:

- Integrated TX/RX switch
- RX path made up of the low noise amplifier (LNA), mixer, low-pass filter (LPF), and variable gain amplifier (VGA)
- TX path made up of the LPF, up converter (UPC), and power amplifier (PA)
- PLL/Local oscillator (LO) path used to provide LO signals for signal paths

Figure 4-1 RF circuit module architecture



4.1.2 Features

The Wi-Fi RF IP has the following features:

- Offers stable LO signals and supports up-conversion and down-conversion of RX and TX signals.
- Supports various calibration functions RX direct current (DC) calibration, TX LO leakage calibration, TX power calibration, and TRX IQ calibration.

4.1.3 Operating Mode

The Wi-Fi RF IP can work in power-on calibration mode, TX mode, RX mode, or sleep mode. The power-on calibration is controlled by the software to compensate the RF circuit performance. In service mode, the PHY controls the TX, RX, or sleep mode of the RF IP.

4.1.4 RF Performance

NOTE

The chip integrates a 2.4G Wi-Fi transceiver. Unless otherwise specified, the performance is described based on the test result of the RF input/output interface **WL_RF_RFIO_2G** of the chip.

Table 4-1, Table 4-2, and Table 4-3 describe the WLAN 2.4 GHz RX and TX performance.

Table 4-1 WLAN 2.4 GHz RX performance

Parameter	Rate	Performance			Unit
		Min.	Typ.	Max.	
Operating frequency band	-	2400	-	2480	MHz
RX sensitivity 11b (8% PER for 1024 octet PSDU)	1 Mbit/s	-	–100	-	dBm
	2 Mbit/s	-	–96.5	-	dBm
	5.5 Mbit/s	-	–95	-	dBm
	11 Mbit/s	-	–92	-	dBm
RX sensitivity 11g (10% PER for 1024 octet PSDU)	6 Mbit/s	-	–96	-	dBm
	9 Mbit/s	-	–94	-	dBm
	12 Mbit/s	-	–93	-	dBm
	18 Mbit/s	-	–91	-	dBm
	24 Mbit/s	-	–88	-	dBm
	36 Mbit/s	-	–85	-	dBm
	48 Mbit/s	-	–81	-	dBm
	54 Mbit/s	-	–79	-	dBm
RX sensitivity 11n 20 MHz HT-MF, 800 ns GI (10% PER for 1024octet PSDU): non-	HT20 MCS0	-	–95	-	dBm
	HT20 MCS1	-	–92	-	dBm

Parameter	Rate	Performance			Unit
		Min.	Typ.	Max.	
STBC	HT20 MCS2	-	–90	-	dBm
	HT20 MCS3	-	–87	-	dBm
	HT20 MCS4	-	–84	-	dBm
	HT20 MCS5	-	–80	-	dBm
	HT20 MCS6	-	–78	-	dBm
	HT20 MCS7	-	–74	-	dBm
Maximum input power	(11b 1M/2M) < 8% PER	-	10	-	dBm
	(11b 5.5M/11M) < 8% PER	-	10	-	dBm
	(11g 6M–54M) < 10% PER	-	TBD	-	dBm
	(11n mcs 0–7) < 10% PER	-	TBD	-	dBm

Table 4-2 WLAN 2.4 GHz RX performance (continued)

Parameter	Rate	Valid Signal Strength (dBm)	Performance			Unit
			Min.	Typ.	Max.	
ACI suppression-11b Ratio of interference signals to useful signals	1 Mbit/s	–74	-	51	-	dB
	2 Mbit/s	–74	-	51	-	dB
	5.5 Mbit/s	–70	-	43	-	dB
	11 Mbit/s	–70	-	42	-	dB
ACI suppression-11g Ratio of interference signals to useful signals	6 Mbit/s	–79	-	44	-	dB
	9 Mbit/s	–78	-	42	-	dB
	12 Mbit/s	–76	-	41	-	dB
	18 Mbit/s	–74	-	40	-	dB
	24 Mbit/s	–71	-	37	-	dB
	36 Mbit/s	–67	-	35	-	dB
	48 Mbit/s	–63	-	31	-	dB
	54 Mbit/s	–62	-	29	-	dB
ACI suppression-11n Ratio of interference signals to useful signals	HT20 MCS0	–79	-	42	-	dB
	HT20 MCS1	–76	-	40	-	dB
	HT20 MCS2	–74	-	39	-	dB

Parameter	Rate	Valid Signal Strength (dBm)	Performance			Unit
			Min.	Typ.	Max.	
	HT20 MCS3	–71	-	37	-	dB
	HT20 MCS4	–67	-	34	-	dB
	HT20 MCS5	–63	-	30	-	dB
	HT20 MCS6	–62	-	29	-	dB
	HT20 MCS7	–61	-	27	-	dB

Table 4-3 WLAN 2.4 GHz TX performance

Parameter	Rate	Parameter			Unit
		Min.	Typ.	Max.	
Maximum TX power (VDD = 3.3 V) in compliance with the mask and EVM protocols	11b (1–11 Mbit/s)	-	22	-	dBm
	11g (6 Mbit/s)	-	21	-	dBm
	11g (54 Mbit/s)	-	20	-	dBm
	OFDM, Binary Phase Shift Keying (BPSK) MCS0	-	19	-	dBm
	OFDM, Quadrature Phase Shift Keying (QPSK) MCS1–2	-	19	-	dBm
	OFDM, 16-Quadrature Amplitude Modulation (QAM) MCS3–4	-	19	-	dBm
	OFDM, 64-QAM MCS6	-	19	-	dBm
	OFDM, 64-QAM MCS7	-	19	-	dBm
Output power precision	Voltage standing wave ratio ()VSWR ≤ 2:1	-	TBD	-	dB
Output power resolution	-	-	TBD	-	dB

NOTE

The test condition is VBAT = 3.3 V. Test data under VBAT = 2.3 V is to be determined

4.2 Wi-Fi ABB

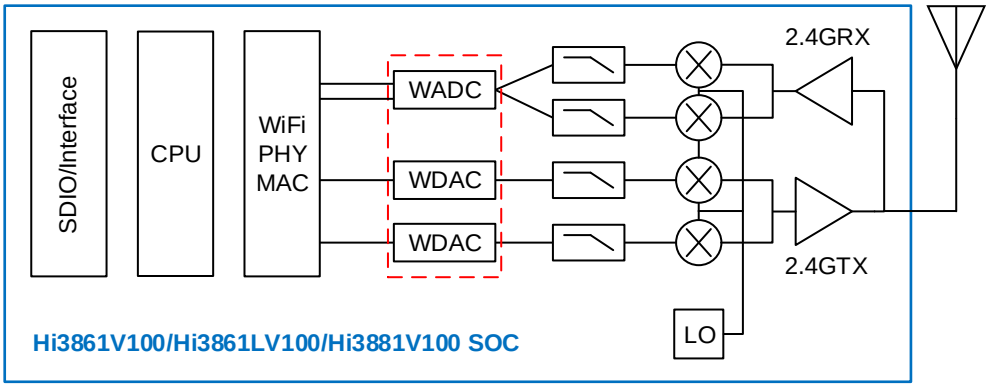
4.2.1 Overview

Applied to the Hi3861/Hi3861L/Hi3881 connectivity SoC, the analog baseband (ABB) IP is the analog digital interface module that supports Wi-Fi 802.11b/g/n (2.4G) and consists of the following two functional modules:

- Wi-Fi IQ-ADC
- Wi-Fi dual-channel DAC

The ABB IP performs digital-to-analog conversion (DAC) for data transmission and analog-to-digital conversion (ADC) upon data receive.

Figure 4-2 ABB IP application in the SoC



NOTE

WADC stands for Wi-Fi analog-to-digital conversion, and WDAC stands for Wi-Fi digital-to-analog conversion.

4.2.2 Features

The ABB IP has the following features:

- One Wi-Fi IQ ADC and one pair of Wi-Fi IQ DACs
- Wi-Fi 802.11b/g/n (2.4G)

4.2.3 Operating Mode

The service mode registers of Hi3861, Hi3861L, and Hi3881 are configured at a time. The registers need to be reconfigured only when the logic power supply is powered off and then powered on. The calibration algorithm is affected by the temperature and voltage drift. If the temperature and voltage change greatly, you need to run the algorithm again and refresh the registers. In other cases, you do not need to reconfigure the registers.

The calibration covers:

- WLAN ADC comparator
- WLAN DC Offset Calibration
- WLAN capacitor

Perform the calibration as follows in sequence:

Step 1 Calibrate the comparator.

Step 2 Calibrate the capacitor.

Step 3 Calibrate the DC offset.

----End

4.3 Wi-Fi PHY

4.3.1 Overview

The WLAN PHY implements the following physical layer functions defined in the 802.11 protocol:

- DSSS and CCK modulation and demodulation defined in 802.11b
- OFDM modulation and demodulation defined in 802.11n and 802.11g, including scrambling, coding (convolutional coding), interleaving, OFDM modulation, Viterbi decoding in the receive direction, and automatic gain control (AGC) and clear channel assessment (CCA) functions
- RF/ABB test and calibration

4.3.2 Features

The Wi-Fi PHY has the following features:

- Supports the IEEE802.11b/g/n wireless local area network (LAN) communication protocol
- Supports the 2.4 GHz band, 5 MHz, 10 MHz, and 20 MHz bandwidths, and a maximum of one stream and one antenna.
- Supports the digital pre-distortion (DPD) and calibration.

4.4 Wi-Fi MAC

4.4.1 Overview

The digital baseband (DBB) MAC implements hardware processing at the Wi-Fi MAC layer, including channel access, framing and deframing, data transmission and reception, encryption and decryption, and energy-saving control.

4.4.2 Features

The Wi-Fi MAC has the following features:

- Supports the IEEE802.11b/g/n wireless local area network (LAN) communication protocol
- Supports the STA and AP modes
- Supports the 2.4 GHz band, 5 MHz, 10 MHz, and 20 MHz bandwidths, and a maximum of one stream and one antenna.

- Supports WPA, WPA2, and AES encryption and decryption, and supports the Chinese National Standard WAPI.
- Supports WPS 2.0.
- Supports protocol-based low power consumption: power saving mode (PSM), unscheduled automatic power save delivery (UAPSD), and peer-to-peer (P2P) power saving.

4.4.3 Operating Modes

The chip supports three basic modes:

- AP mode
- STA mode
- AP/STA coexistence mode

NOTE

The features described in the following sections are derived from the three basic modes.

AP Mode

A basic service set (BSS) network provides basic functions of all access points, including:

- Sends Beacon frames to declare the existence and capability of the BSS.
- Provides wireless association and authentication services for clients in the BSS.
- Manages associated clients on the BSS network.

The chip supports one AP.

STA Mode

On a basic BSS network, the chip scans and discovers networks, connects to networks, and manages connections to APs to provide data transmission and reception services.

The chip supports two STAs.

Monitor Mode

The chip enters the monitor mode to implement the packet capture function of the network adapter. The MAC reports all received frames to the software.

AP and STA Coexistence

Each of Hi3861, Hi3861L, and Hi3881 offers a single MAC, a single PHY, and two RFs. Therefore, the dual band dual concurrent (DBDC) function cannot be implemented. However, the DBAC function can be implemented by fast switching between frequency bands (that is, time division multiplexing) so that multiple devices can coexist.

The chip allows one AP and one STA to work at the same time.

The chip supports concurrent access of APs and STAs on the same or different channels at the 2.4 GHz frequency band, which corresponds to intra-frequency coexistence and inter-frequency coexistence, respectively.

Constraints: After the STA is powered on, it scans channels, causing channel switching. Therefore, before enabling the AP/STA dynamic coexistence function, you need to create a STA and then an AP. Otherwise, the working channel of the AP is affected.

CSI Mode

In CSI mode, the channel state information (CSI) reported by the PHY can be filtered and then reported to software. The following features are supported:

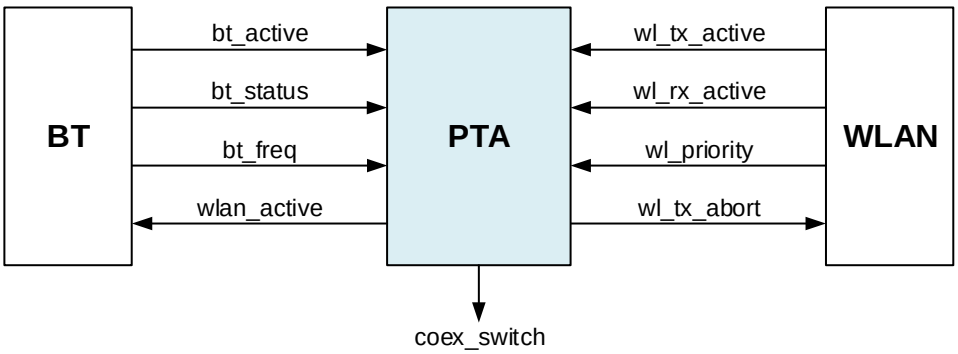
- Supports reporting of 11g/11n CSI information, instead of 11b CSI information.
- Supports source address filtering for the extracted CSI frames. There are six source address filtering lists (whitelists). The associated device uses the address content in the lookup table (LUT).
- Supports six CSI sampling periods. The CSI sampling period is bound to the whitelists. One whitelist corresponds to one collection period.
- Supports reporting of CSI information based on filter criteria, which can be the whitelist, sampling period, and specific frame type.
- Reports the H matrix in I/Q format with 12-bit precision.
- Supports reporting of the bandwidth (20 MHz), frame format (NON-HT or HT-MF), received signal strength indicator (RSSI), and signal noise ratio (SNR) along with the CSI information (STBC frame reporting is not supported). L-LTF H matrix data can be reported. Fine timing measurement (FTM) is not supported.

4.5 Packet Traffic Arbitration (PTA)

4.5.1 Overview

PTA is the arbitration module for BT and WLAN communication packets. It provides a group of BT request interfaces and a group of WLAN request interfaces to identify the TX and RX status and priority of requests and return the arbitration result. When the BT and WLAN share the same antenna, the coex_switch signal is output based on the arbitration result, instructing the antenna to switch between the BT and WLAN.

Figure 4-3 PTA block diagram



4.5.2 Features

BT Request Interface

The BT request interface has the following features:

- Provides a group of BT request interfaces to identify BT TX and RX requests and priorities and return the arbitration result.
- Supports four coexistence modes: COEX_MODE_2A, COEX_MODE_2B, COEX_MODE_3, and COEX_MODE_4.
- Supports two arbitration modes: independent antenna and co-antenna.
- Supports the input/output enable and valid level configuration of the BT request interface.
- Allows the arbitration result wlan_active to be controlled by hardware or configured by software (software configuration has a higher priority).

WLAN Request Interface

The WLAN request interface has the following features:

- Provides a group of WLAN request interfaces to identify WLAN TX and RX requests and priorities and return the arbitration result.
- Supports four coexistence modes: COEX_MODE_2A, COEX_MODE_2B, COEX_MODE_3, and COEX_MODE_4.
- Supports two arbitration modes: independent antenna and co-antenna.
- Allows the arbitration result wl_tx_abort to be controlled by hardware or masked by software.

COEX_SWITCH Interface

The COEX_SWITCH interface has the following features:

- Outputs the coex_switch signal to instruct the antenna to switch between Bluetooth and WLAN.
- Supports four coexistence modes: COEX_MODE_2A, COEX_MODE_2B, COEX_MODE_3, and COEX_MODE_4.
- Applies only to co-antenna scenarios.
- Allows the coex_switch signal of the antenna arbitration result to be controlled by hardware or configured by software (software configuration has a higher priority).

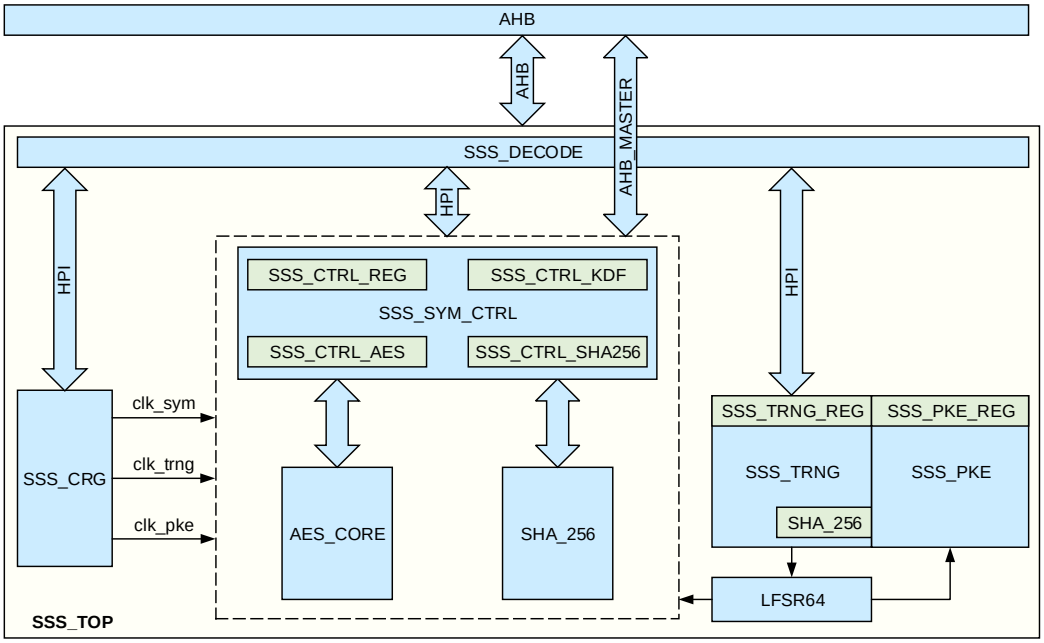
5 Security System

5.1 Security Subsystem

5.1.1 Overview

The security subsystem (SSS) consists of a range of algorithm modules including the Advanced Encryption Standard (AES), HASH, key derivation function (KDF), public key engine (PKE), and True Random Number Generator (TRNG) modules. The SSS is connected to the CPU through the advanced high-performance bus (AHB). Figure 5-1 shows the logical block diagram of the SSS. The CPU invokes the AES, HASH, KDF, PKE, and TRNG operators of the SSS through the AHB. In addition, the CPU can control the clock gating for a specific operator through the corresponding register to reduce power consumption.

Figure 5-1 SSS logical block diagram



Note: HPI is short for the hardware platform interface.

5.1.2 Features

The SSS supports the following algorithms:

- AES
- HASH
- KDF
- PKE
- TRNG

5.2 AES

5.2.1 Overview

The AES module implements the AES algorithm for data encryption and decryption. Software configures the key, initial value, working mode, encryption/decryption, key length, length of the data to be processed, address of the data to be processed, whether to return the processing result, and write back address, and then starts AES computation.

The AES algorithm complies with the Federal Information Processing Standard (FIPS) 197.

5.2.2 Features

The AES module has the following features:

- AES encryption and decryption
- ECB, CBC, and CTR modes
- 128-bit, 192-bit, and 256-bit key lengths
- Reading data to be processed in DMA mode
- Data lengths up to 4096 bytes. The configured data length must be aligned by block, that is, a multiple of 16 bytes.
- Software-configured or KDF-derived key
- Result return: In CBC mode, software controls whether the result is returned in DMA mode by configuring registers. If the write back operation is no enabled, software can read logical register to obtain the final result.
- Random delay for starting computation

5.3 HASH

5.3.1 Overview

The HASH module implements the SHA256 algorithm for data integrity authentication. The software enables or disables initial value update (the initial value needs to be configured), and then starts the SHA256 algorithm to compute the message digest.

The SHA256 algorithm complies with the FIPS 180-2.

5.3.2 Features

The HASH module has the following features:

- SHA256 algorithm
- Reading input data in DMA mode
- Data lengths up to 4096 bytes. The configured data length must be aligned by block, that is, a multiple of 64 bytes. The configured data length is the length after the data is padded by software.
- Configurable initial value

5.4 KDF

5.4.1 Overview

The KDF module derives keys based on the input.

The KDF computation complies with Password-Based Key Derivation Function 2 (PBKDF2) defined in SP 800-132.

5.4.2 Features

The KDF module has the following features:

- KDF implementation by invoking HMAC-SHA256
- Compliance with PBKDF2, SP 800-132
- Configurable iterations for the KDF to call HMAC-SHA256
- Device key derivation and stored key derivation
- Allowance of revoking the CPU permission to read the key stored in the eFUSE

5.5 PKE

5.5.1 Overview

The PKE module implements asymmetric cryptographic algorithms Rivest-Shamir-Adleman (RSA) and elliptic curve cryptography (ECC). The PKE module communicates with the external controller through the AHB interface. The external controller can access the internal registers of PKE module through the AHB, by configuring the computation parameters, reading the computation result, and querying the status of the PKE module.

5.5.2 Features

The PKE has the following features:

- Two asymmetric algorithms: RSA and ECC
- Key bit widths: 2048 bits, 3072 bits, and 4096 bits for RSA; 256 bits for ECC
- Basic operators, including modular inverse, modular multiplication, modulo, large-number multiplication, modular addition, and modular subtraction.

5.6 TRNG

5.6.1 Overview

The TRNG generates true random numbers.

5.6.2 Features

The TRNG has the following features:

- True random number generation
- Eight ring oscillators (ROs) as random sources
- Deterministic random bit generator (DRBG), complying with SP 800-90A
- Generated random number performance > 4Mbit/s at 24 MHz RO sampling clock and at 160 MHz system clock

5.7 eFUSE

5.7.1 Overview

The eFUSE is a one-time programmable (OTP) memory. It is mainly used to store the chip ID, key, or other one-off data of the chip.

5.7.2 Features

The eFUSE has the following features:

- 2048-bit storage capacity. Based on the read and write permissions of the software, there are four areas:
  - Area readable and writable by software
  - Area unreadable by software
  - Area readable by software only when the CPU interrupt condition is met
  - Area writable by software only when the CPU interrupt condition is met

  Table 5-1 describes the eFUSE bit assignments.

Table 5-1 eFUSE bit assignments

Bit Field	Bit Width	Readable by Software	Writable by Software
bit[751:496]	256	No	Yes
bit[1671:1656]	16	Yes	CPU interrupt control
bit[1719:1672]	48	Yes	
bit[1847:1720]	128	CPU interrupt control	
bit[2036]	1	Yes	
bit[2037]	1	Yes	

Bit Field	Bit Width	Readable by Software	Writable by Software
bit[2038]	1	Yes	
Other	-	Yes	Yes

- Area bit[2047:0] consists of one locked area and 41 subareas.
  - Bit[2047:2012]: lock bits of areas 35–0 in sequence
  - Bit[239:235]: lock bits of areas 40–36 in sequence

  If the lock bit of area 31 is programed to **1**, the lock function takes effect immediately and the software cannot program the area any more. If the lock bit of a different area is programed to **1**, the lock function takes effect only after the system is powered on again, and the software cannot program this area any more.

  Table 5-2 shows the detailed eFUSE bit assignments.

Table 5-2 eFUSE bit assignments

Field	Description	Bit Field	Bit Count	Lock Bit	Write Source	Readable by DFT	Writable by DFT	Readable by Software	Writable by Software
die_id	DIE ID	bit[199:8]	192	PG1	ATE	1	1	1	1
flash_encpt_cnt[7:6]	Firmware encryption/decryption count 4	bit[221:220]	2	PG36	SW	1	1	1	1
flash_encpt_cnt[9:8]	Firmware encryption/decryption count 5	bit[223:222]	2	PG37	SW	1	1	1	1
flash_encpt_cnt[11:10]	Firmware encryption/decryption count 6	bit[225:224]	2	PG38	SW	1	1	1	1
rsvd0	-	bit[233:226]	8	PG39	SW	1	1	1	1
PG36	Lock bit	bit[235]	1	-	SW	1	1	1	1
PG37	Lock bit	bit[236]	1	-	SW	1	1	1	1
PG38	Lock bit	bit[237]	1	-	SW	1	1	1	1
PG39	Lock bit	bit[238]	1	-	SW	1	1	1	1
PG40	Lock bit	bit[239]	1	-	SW	1	1	1	1
root_pubkey	Hash value of the root public key	bit[495:240]	256	PG3	SW	0	1	1	1

Field	Description	Bit Field	Bit Count	Lock Bit	Write Source	Readable by DFT	Writable by DFT	Readable by Software	Writable by Software
root_key	Root key	bit[751:496]	256	PG4	SW	0	1	0	1
uart_pubkey	Hash value of the UART public key for verification	bit[1007:752]	256	PG5	SW	0	1	1	1
subkey_cat	Level-2 key type	bit[1039:1008]	32	PG6	SW	1	1	1	1
ivn	Level-2 key revocation flag	bit[1071:1040]	32	PG7	SW	1	1	1	1
jtm	JTAG interface mask	bit[1073]	1	PG8	SW	1	1	1	1
utm0	UART0 interface mask	bit[1074]	1	PG9	SW	1	1	1	1
utm1	UART1 interface mask	bit[1075]	1	PG10	SW	1	1	1	1
utm2	UART2 interface mask	bit[1076]	1	PG11	SW	1	1	1	1
sdc	Secure debugging control	bit[1077]	1	PG12	SW	1	1	1	1
kdf2ecc_huk_disable	KDF key output return disable signal with the hardware unique key (HUK)	bit[1079]	1	PG35	ATE	1	1	1	1
boot_uart_sel	DBG UART interface select during boot	bit[1081:1080]	2	PG14	SW	1	1	1	1
uart_halt_interval	DBG UART interrupt wait time during boot	bit[1083:1082]	2	PG15	SW	1	1	1	1
ts_trim	T-Sensor temperature trimming code	bit[1087:1084]	4	PG2	ATE	1	1	1	1
abb_trim	ABB internal register trimming value (1088-bit): 0: Non-SSS corner 1: SSS corner	bit[1095:1088]	8	PG16	ATE	1	1	1	1
ipv4_mac_addr	IPv4 MAC address	bit[1143:1096]	48	PG17	SW	1	1	1	1
ipv6_mac_addr	IPv6 MAC address	bit[1271:1144]	128	PG17	SW	1	1	1	1

Field	Description	Bit Field	Bit Count	Lock Bit	Write Source	Readable by DFT	Writable by DFT	Readable by Software	Writable by Software
pa2gccka0_trim0	802.11b core 0 power trimming factor, round 1	bit[1303:1272]	32	PG18	SW	1	1	1	1
pa2gccka1_trim0	802.11b core 0 power trimming factor, round 1	bit[1335:1304]	32	PG18	SW	1	1	1	1
nvram_pa2ga0_trim0	802.11g 20 MHz power trimming factor, round 1	bit[1367:1336]	32	PG19	SW	1	1	1	1
nvram_pa2ga1_trim0	802.11g 20 MHz power trimming factor, round 1	bit[1399:1368]	32	PG19	SW	1	1	1	1
pa2gccka0_trim1	802.11b core 0 power trimming factor, round 2	bit[1431:1400]	32	PG20	SW	1	1	1	1
pa2gccka1_trim1	802.11b core 0 power trimming factor, round 2	bit[1463:1432]	32	PG20	SW	1	1	1	1
nvram_pa2ga0_trim1	802.11g 20 MHz power trimming factor, round 2	bit[1495:1464]	32	PG21	SW	1	1	1	1
nvram_pa2ga1_trim1	802.11g 20 MHz power trimming factor, round 2	bit[1527:1496]	32	PG21	SW	1	1	1	1
pa2gccka0_trim2	802.11b core 0 power trimming factor, round 3	bit[1559:1528]	32	PG22	SW	1	1	1	1
pa2gccka1_trim2	802.11b core 0 power trimming factor, round 3	bit[1591:1560]	32	PG22	SW	1	1	1	1
nvram_pa2ga0_trim2	802.11g 20 MHz power trimming factor, round 3	bit[1623:1592]	32	PG23	SW	1	1	1	1
nvram_pa2ga1_trim2	802.11g 20 MHz power trimming factor, round 3	bit[1655:1624]	32	PG23	SW	1	1	1	1
tee_boot_ver	TEE boot version	bit[1671:1656]	16	PG24	SW	1	0	1	1*

Field	Description	Bit Field	Bit Count	Lock Bit	Write Source	Readable by DFT	Writable by DFT	Readable by Software	Writable by Software
tee_firmware_ver	TEE firmware version	bit[1719:1672]	48	PG25	SW	1	0	1	1*
tee_salt	TEE salt	bit[1847:1720]	128	PG26	SW	0	0	1*	1*
flash_encpt_cnt[1:0]	Firmware encryption/decryption count 1	bit[1849:1848]	2	PG27	SW	1	1	1	1
flash_encpt_cnt[3:2]	Firmware encryption/decryption count 2	bit[1851:1850]	2	PG28	SW	1	1	1	1
flash_encpt_cnt[5:4]	Firmware encryption/decryption count 3	bit[1853:1852]	2	PG29	SW	1	1	1	1
flash_encpt_cfg	Firmware encryption/decryption control	bit[1854]	1	PG30	SW	1	1	1	1
flash_scramble_en	Flash data scrambling enable	bit[1855]	1	PG31	SW	1	1	1	1
user_flash_ind	Salt for flash data scrambling	bit[1865:1856]	10	PG31	SW	1	1	1	1
rf_pdbuffer_gcal	RF trimming factor	bit[1883:1866]	18	PG32	ATE	1	1	1	1
customer_rsvd0	Reserved for the user	bit[1947:1884]	64	PG33	SW	1	1	1	1
customer_rsvd1	Reserved for the user	bit[2011:1948]	64	PG34	SW	1	1	1	1
PG0	Lock bit	bit[2012]	1	-	ATE	1	1	1	1
PG1	Lock bit	bit[2013]	1	-	ATE	1	1	1	1
PG2	Lock bit	bit[2014]	1	-	ATE	1	1	1	1
PG3	Lock bit	bit[2015]	1	-	ATE	1	1	1	1
PG4	Lock bit	bit[2016]	1	-	SW	1	1	1	1
PG5	Lock bit	bit[2017]	1	-	SW	1	1	1	1
PG6	Lock bit	bit[2018]	1	-	SW	1	1	1	1

Field	Description	Bit Field	Bit Count	Lock Bit	Write Source	Readable by DFT	Writable by DFT	Readable by Software	Writable by Software
PG7	Lock bit	bit[2019]	1	-	SW	1	1	1	1
PG8	Lock bit	bit[2020]	1	-	SW	1	1	1	1
PG9	Lock bit	bit[2021]	1	-	SW	1	1	1	1
PG10	Lock bit	bit[2022]	1	-	SW	1	1	1	1
PG11	Lock bit	bit[2023]	1	-	SW	1	1	1	1
PG12	Lock bit	bit[2024]	1	-	SW	1	1	1	1
PG13	Lock bit	bit[2025]	1	-	SW	1	1	1	1
PG14	Lock bit	bit[2026]	1	-	SW	1	1	1	1
PG15	Lock bit	bit[2027]	1	-	SW	1	1	1	1
PG16	Lock bit	bit[2028]	1	-	SW	1	1	1	1
PG17	Lock bit	bit[2029]	1	-	SW	1	1	1	1
PG18	Lock bit	bit[2030]	1	-	SW	1	1	1	1
PG19	Lock bit	bit[2031]	1	-	SW	1	1	1	1
PG20	Lock bit	bit[2032]	1	-	SW	1	1	1	1
PG21	Lock bit	bit[2033]	1	-	SW	1	1	1	1
PG22	Lock bit	bit[2034]	1	-	SW	1	1	1	1
PG23	Lock bit	bit[2035]	1	-	SW	1	1	1	1
PG24	Lock bit	bit[2036]	1	-	SW	1	0	1	1*
PG25	Lock bit	bit[2037]	1	-	SW	1	0	1	1*
PG26	Lock bit	bit[2038]	1	-	SW	1	0	1	1*
PG27	Lock bit	bit[2039]	1	-	SW	1	1	1	1
PG28	Lock bit	bit[2040]	1	-	SW	1	1	1	1
PG29	Lock bit	bit[2041]	1	-	SW	1	1	1	1
PG30	Lock bit	bit[2042]	1	-	SW	1	1	1	1
PG31	Lock bit	bit[2043]	1	-	SW	1	1	1	1
PG32	Lock bit	bit[2044]	1	-	SW	1	1	1	1

Field	Description	Bit Field	Bit Count	Lock Bit	Write Source	Readable by DFT	Writable by DFT	Readable by Software	Writable by Software
PG33	Lock bit	bit[2045]	1	-	SW	1	1	1	1
PG34	Lock bit	bit[2046]	1	-	SW	1	1	1	1
PG35	Lock bit	bit[2047]	1	-	SW	1	1	1	1

6 Peripherals

6.1 I/O MUX

6.1.1 Overview

The number of digital pins is limited, we need to enrich the pin functions by I/O multiplexing.

6.1.2 Description of Software Multiplexed Pins

NOTICE

- Real-time clock (RTC): Table 6-1 shows the RTC solution of Hi3861, Hi3861L, and Hi3881. Hi3861 and Hi3881 have no external RTC clock. GPIO_00 and GPIO_01 are used for GPIO and can be multiplexed with I/O. Hi3861L has an external RTC. When the crystal clock solution is used, GPIO_00 functions as the RTC32K_XOUT pin and GPIO_01 functions as the RTC32K_XIN pin. When the crystal oscillator clock solution is used, GPIO_00 functions as the RTC_OSC_32K pin, and GPIO_01 functions as the GPIO function for I/O multiplexing.
- Analog to Digital Converter (ADC) pins: Only the LSADC channel or the GPIO function is supported. Table 6-2 describes the mapping between ADC channel pins and GPIO pins.

Table 6-1 RTC clock solution

Pad Info	Hi3861	Hi3861L		Hi3881
		Crystal	Crystal Oscillator	
GPIO_00	GPIO function, I/O multiplexing supported	RTC32K_XOUT	RTC_OSC_32K	GPIO function, I/O multiplexing supported
GPIO_01	GPIO function, I/O multiplexing supported	RTC32K_XIN	GPIO function, I/O multiplexing supported	GPIO function, I/O multiplexing supported

Table 6-2 Mapping between ADC channel pins and multiplexed pins

Multiplexed Pin	ADC Pin
GPIO_04	ADC1
GPIO_05	ADC2
GPIO_07	ADC3
GPIO_09	ADC4
GPIO_11	ADC5
GPIO_12	ADC0
GPIO_13	ADC6

Table 6-3 describes the software multiplexed pins.

Table 6-3 Software multiplexed pins

Pin No.	Pad Name	Multiplex Control Register	Multiplexed Signal 0	Multiplexed Signal 1	Multiplexed Signal 2	Multiplexed Signal 3	Multiplexed Signal 4	Multiplexed Signal 5	Multiplexed Signal 6	Multiplexed Signal 7
2	GPIO_00	GPIO_00_SEL	GPIO[0]	HW_ID[0]	UART1_TXD	SPI1_CK	JTAG_TDO	PWM3_OUT	I2C1_SDA	-
3	GPIO_01	GPIO_01_SEL	GPIO[1]	HW_ID[1]	UART1_RXD	SPI1_RXD	JTAG_TCK	PWM4_OUT	I2C1_SCL	BT_FREQ
4	GPIO_02	GPIO_02_SEL	GPIO[2]	REFCLK_FREQ_STATUS	UART1_RTS_N	SPI1_TXD	JTAG_TRSTN	PWM2_OUT	DIAG[0]	SSI_CLK
5	GPIO_03	GPIO_03_SEL	GPIO[3]	UART0_TXD	UART1_CTS_N	SPI1_CSN	JTAG_TDI	PWM5_OUT	I2C1_SDA	SSI_DATA
6	GPIO_04	GPIO_04_SEL	GPIO[4]	HW_ID[3]	UART0_RXD	JTAG_TMS	PWM1_OUT	I2C1_SCL	DIAG[7]	-
17	GPIO_05	GPIO_05_SEL	GPIO[5]	HW_ID[4]	UART1_RXD	SPI0_CSN	DIAG[1]	PWM2_OUT	I2S0_MCLK	BT_STATUS
18	GPIO_06	GPIO_06_SEL	GPIO[6]	JTAG_MODE	UART1_TXD	SPI0_CK	DIAG[2]	PWM3_OUT	I2S0_TX	COEX_SWITCH
19	GPIO_07	GPIO_07_SEL	GPIO[7]	HW_ID[5]	UART1_CTS_N	SPI0_RXD	DIAG[3]	PWM0_OUT	I2S0_BCLK	BT_ACTIVE
20	GPIO_08	GPIO_08_SEL	GPIO[8]	JTAG_ENABLE	UART1_RTS_N	SPI0_TXD	DIAG[4]	PWM1_OUT	I2S0_WS	WLAN_ACTIVE
27	GPIO_09	GPIO_09_SEL	GPIO[9]	I2C0_SCL	UART2_RTS_N	SDIO_D2	SPI0_TXD	PWM0_OUT	DIAG[5]	I2S0_MCLK
28	GPIO_10	GPIO_10_SEL	GPIO[10]	I2C0_SDA	UART2_CTS_N	SDIO_D3	SPI0_CK	PWM1_OUT	DIAG[6]	I2S0_TX
29	GPIO_11	GPIO_11_SEL	GPIO[11]	HW_ID[6]	UART2_TXD	SDIO_CMD	SPI0_RXD	PWM2_OUT	RF_TX_EN_EXT	I2S0_RX
30	GPIO_12	GPIO_12_SEL	GPIO[12]	HW_ID[7]	UART2_RXD	SDIO_CLK	SPI0_CSN	PWM3_OUT	RF_RX_EN_EXT	I2S0_BCLK
31	GPIO_13	GPIO_13_SEL	SSI_DATA	UART0_TXD	UART2_RTS_N	SDIO_D0	GPIO[13]	PWM4_OUT	I2C0_SDA	I2S0_WS

Pin No.	Pad Name	Multiplex Control Register	Multiplexed Signal 0	Multiplexed Signal 1	Multiplexed Signal 2	Multiplexed Signal 3	Multiplexed Signal 4	Multiplexed Signal 5	Multiplexed Signal 6	Multiplexed Signal 7
32	GPIO_14	GPIO_14_SEL	SSI_CLK	UART0_RXD	UART2_CTS_N	SDIO_D1	GPIO[14]	PWM5_OUT	I2C0_SCL	HW_ID[2]
27	SFC_CSN	SFC_CSN_SEL	SFC_CSN	SDIO_D2	GPIO[9]	DIAG[5]	SPI0_TXD	-	-	-
28	SFC_IO1	SFC_IO1_SEL	SFC_DO	SDIO_D3	GPIO[10]	DIAG[6]	SPI0_CK	-	-	-
29	SFC_IO2	SFC_IO2_SEL	SFC_WPN	SDIO_CMD	GPIO[11]	RF_TX_EN_EXT	SPI0_RXD	-	-	-
30	SFC_IO0	SFC_IO0_SEL	SFC_DI	SDIO_CLK	GPIO[12]	RF_RX_EN_EXT	SPI0_CSN	-	-	-
31	SFC_CLK	SFC_CLK_SEL	SFC_CLK	SDIO_D0	GPIO[13]	SSI_DATA	-	-	-	-
32	SFC_IO3	SFC_IO3_SEL	SFC_HOLDN	SDIO_D1	GPIO[14]	SSI_CLK	-	-	-	-

NOTE

The package functions of pins in red vary with chips. For details about the differences, see the following NOTICE.

NOTICE

- For Hi3881, SFC_CSN, SFC_IO1, SFC_IO2, SFC_IO0, SFC_CLK, and SFC_IO3 are used as pins 27 to 32, and pins GPIO_09 to GPIO_14 are not outputs.
- Hi3881 supports the UART2, I2C0, HW_ID2, HW_ID6, or HW_ID7 functions.
- Do not configure the multiplexing functions of pins GPIO_09 to GPIO_14 of Hi3881 as the same as those of SFC_CSN, SFC_IO1, SFC_IO2, SFC_IO0, SFC_CLK, and SFC_IO3. Otherwise, the SFC pins cannot be used when they are multiplexed as non-SFC functions. If GPIO_09_SEL is set to **3** (as SDIO function), and SFC_CSN_SEL is set to **1**, the SFC_CSN pin cannot be used as the SDIO function. Therefore, you are advised to set GPIO_09_SEL–GPIO_14_SEL to **0x2** or **0x7**.
- GPIO_09 to GPIO_14 of Hi3861 and Hi3861L are used as pins 27 to 32, and SFC_CSN, SFC_IO1, SFC_IO2, SFC_IO0, SFC_CLK, and SFC_IO3 are not outputs.
- For Hi3861 and Hi3861L, set pins SFC_CSN, SFC_IO1, SFC_IO2, SFC_IO0, SFC_CLK, and SFC_IO3 as the SFC function. Otherwise, the embedded flash memory is unavailable.

Table 6-4 lists the software multiplexed pins of the GPIO.

Table 6-4 Description of the software multiplexed pins of the GPIO

Signal	Direction	Description
BT_ACTIVE	I	BT service request signal
BT_FREQ	I	BT channel status signal
BT_STATUS	I	BT service status signal
COEX_SWITCH	O	Antenna switching indicator
DIAG[0]	O	Maintenance and test signal
DIAG[1]	O	Maintenance and test signal
DIAG[2]	O	Maintenance and test signal
DIAG[3]	O	Maintenance and test signal
DIAG[4]	O	Maintenance and test signal
DIAG[5]	O	Maintenance and test signal
DIAG[6]	O	Maintenance and test signal
DIAG[7]	O	Maintenance and test signal
GPIO[0]	I/O	GPIO pin signal
GPIO[1]	I/O	GPIO pin signal
GPIO[10]	I/O	GPIO pin signal
GPIO[11]	I/O	GPIO pin signal
GPIO[12]	I/O	GPIO pin signal

Signal	Direction	Description
GPIO[13]	I/O	GPIO pin signal
GPIO[14]	I/O	GPIO pin signal
GPIO[2]	I/O	GPIO pin signal
GPIO[3]	I/O	GPIO pin signal
GPIO[4]	I/O	GPIO pin signal
GPIO[5]	I/O	GPIO pin signal
GPIO[6]	I/O	GPIO pin signal
GPIO[7]	I/O	GPIO pin signal
GPIO[8]	I/O	GPIO pin signal
GPIO[9]	I/O	GPIO pin signal
HW_ID[0]	I	Hardware control word
HW_ID[1]	I	Hardware control word
HW_ID[2]	I	Hardware control word
HW_ID[3]	I	Hardware control word
HW_ID[4]	I	Hardware control word
HW_ID[5]	I	Hardware control word
HW_ID[6]	I	Hardware control word
HW_ID[7]	I	Hardware control word
I2C0_SCL	I/O	I^2C clock
I2C0_SDA	I/O	I^2C data /nSET
I2C1_SCL	I/O	I^2C clock
I2C1_SDA	I/O	I^2C data /nSET
I2S0_BCLK	O	I^2S working clock signal
I2S0_MCLK	O	I^2S master clock signal
I2S0_RX	I	I^2S data RX signal
I2S0_TX	O	I^2S data TX signal
I2S0_WS	O	I^2S channel select signal
JTAG_ENABLE	I	JTAG enable 0: common I/O 1: JTAG enabled

Signal	Direction	Description
JTAG_MODE	I	JTAG mode select signal 0: normal mode 1: design for testability (DFT) test mode
JTAG_TCK	I	JTAG clock input
JTAG_TDI	I	JTAG data input
JTAG_TDO	I/O	JTAG data output
JTAG_TMS	I/O	JTAG mode select input
JTAG_TRSTN	I	JTAG reset input, active low. By default, this signal is in reset state.
PWM0_OUT	O	PWM0 output signal
PWM1_OUT	O	PWM1 output signal
PWM2_OUT	O	PWM2 output signal
PWM3_OUT	O	PWM3 output signal
PWM4_OUT	O	PWM4 output signal
PWM5_OUT	O	PWM5 output signal
REFCLK_FREQ_STATUS	I	Crystal clock frequency indicator 0: 40 MHz 1: 24 MHz
RF_RX_EN_EXT	I	Externally input RF RX enable signal
RF_TX_EN_EXT	I	Externally input RF TX enable signal
SDIO_CLK	I	SDIO clock signal
SDIO_CMD	I/O	SDIO command signal
SDIO_D0	I/O	SDIO data signal 0
SDIO_D1	I/O	SDIO data signal 1
SDIO_D2	I/O	SDIO data signal 2
SDIO_D3	I/O	SDIO data signal 3
SPI0_CK	I/O	SPI0 clock signal
SPI0_CSN	I/O	SPI0 chip select (CS) signal
SPI0_RXD	I	SPI0 data RX signal
SPI0_TXD	I/O	SPI0 data TX signal
SPI1_CK	I/O	SPI1 clock signal
SPI1_CSN	I/O	SPI1 chip select (CS) signal

Signal	Direction	Description
SPI1_RXD	I	SPI1 data RX signal
SPI1_TXD	I/O	SPI1 data TX signal
SSI_CLK	I	SSI clock signal
SSI_DATA	I/O	SSI data signal
UART0_RXD	I	Main board communication UART RX
UART0_TXD	O	Main board communication UART TX
UART1_CTS_N	I	UART1 flow control signal
UART1_RTS_N	O	UART1 flow control signal
UART1_RXD	I	Debugging UART1 RX
UART1_TXD	O	Debugging UART1 TX
UART2_CTS_N	I	UART2 flow control signal
UART2_RTS_N	O	UART2 flow control signal
UART2_RXD	I	Debugging UART2 RX
UART2_TXD	O	Debugging UART2 TX
WLAN_ACTIVE	O	WLAN service status indicator
SFC_CLK	O	Flash control signal Flash clock range: The phase-locked loop (PLL) of the central monitor unit (CMU) generates clocks of 96 MHz, 80 MHz, 60 MHz, and 48 MHz. The crystal clock (20 MHz or 12 MHz) is used during power-on.
SFC_CSN	O	Flash CS signal
SFC_DI	I/O	Flash data signal 0
SFC_DO	I/O	Flash data signal 1
SFC_HOLDN	I/O	Flash data signal 3
SFC_WPN	I/O	Flash data signal 2

6.1.3 Summary of Multiplex Control Registers

Table 6-5 describes the pin multiplexing registers.

Table 6-5 Summary of multiplexing registers (base address: 0x5000_A000)

Offset Address	Description	Description	Page
0x604	GPIO_00_SEL	GPIO_00 pin multiplex control register	6-10
0x608	GPIO_01_SEL	GPIO_01 pin multiplex control register	6-10
0x60C	GPIO_02_SEL	GPIO_02 pin multiplex control register	6-11
0x610	GPIO_03_SEL	GPIO_03 pin multiplex control register	6-11
0x614	GPIO_04_SEL	GPIO_04 pin multiplex control register	6-12
0x618	GPIO_05_SEL	GPIO_05 pin multiplex control register	6-12
0x61C	GPIO_06_SEL	GPIO_06 pin multiplex control register	6-13
0x620	GPIO_07_SEL	GPIO_07 pin multiplex control register	6-13
0x624	GPIO_08_SEL	GPIO_08 pin multiplex control register	6-14
0x628	GPIO_09_SEL	GPIO_09 pin multiplex control register	6-14
0x62C	GPIO_10_SEL	GPIO_10 pin multiplex control register	6-15
0x630	GPIO_11_SEL	GPIO_11 pin multiplex control register	6-15
0x634	GPIO_12_SEL	GPIO_12 pin multiplex control register	6-16
0x638	GPIO_13_SEL	GPIO_13 pin multiplex control register	6-16
0x63C	GPIO_14_SEL	GPIO_14 pin multiplex control register	6-17
0x640	SFC_CSN_SEL	SFC_CSN pin multiplex control register	6-17
0x644	SFC_IO1_SEL	SFC_IO1 pin multiplex control register	6-18
0x648	SFC_IO2_SEL	SFC_IO2 pin multiplex control register	6-18
0x64C	SFC_IO0_SEL	SFC_IO0 pin multiplex control register	6-19
0x650	SFC_CLK_SEL	SFC_CLK pin multiplex control register	6-19
0x654	SFC_IO3_SEL	SFC_IO3 pin multiplex control register	6-20

6.1.4 Description of Multiplexing Registers

GPIO_00_SEL

GPIO_00_SEL is the multiplex control register for the GPIO_00 pin.

Offset Address: 0x604 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_00_sel	GPIO_00 pin multiplex control 000: GPIO[0] 001: HW_ID[0] 010: UART1_TXD 011: SPI1_CK 100: JTAG_TDO 101: PWM3_OUT 110: I2C1_SDA Other values: reserved	0x0

GPIO_01_SEL

GPIO_01_SEL is the multiplex control register of the GPIO_01 pin.

Offset Address: 0x608 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_01_sel	GPIO_01 pin multiplex control 000: GPIO[1] 001: HW_ID[1] 010: UART1_RXD 011: SPI1_RXD 100: JTAG_TCK 101: PWM4_OUT 110: I2C1_SCL 111: BT_FREQ	0x0

GPIO_02_SEL

GPIO_02_SEL is the multiplex control register of the GPIO_02 pin.

Offset Address: 0x60C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_02_sel	GPIO_02 pin multiplex control 000: GPIO[2] 001: REFCLK_FREQ_STATUS 010: UART1_RTS_N 011: SPI1_TXD 100: JTAG_TRSTN 101: PWM2_OUT 110: DIAG[0] 111: SSI_CLK	0x0

GPIO_03_SEL

GPIO_03_SEL is the multiplex control register of the GPIO_03 pin.

Offset Address: 0x610 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_03_sel	GPIO_03 pin multiplex control 000: GPIO[3] 001: UART0_TXD 010: UART1_CTS_N 011: SPI1_CSN 100: JTAG_TDI 101: PWM5_OUT 110: I2C1_SDA 111: SSI_DATA	0x0

GPIO_04_SEL

GPIO_04_SEL is the multiplex control register of the GPIO_04 pin.

Offset Address: 0x614 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_04_sel	GPIO_04 pin multiplex control 000: GPIO[4] 001: HW_ID[3] 010: UART0_RXD 100: JTAG_TMS 101: PWM1_OUT 110: I2C1_SCL 111: DIAG[7] Other values: reserved	0x0

GPIO_05_SEL

GPIO_05_SEL is the multiplex control register of the GPIO_05 pin.

Offset Address: 0x618 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_05_sel	GPIO_05 pin multiplex control 000: GPIO[5] 001: HW_ID[4] 010: UART1_RXD 011: SPI0_CSN 100: DIAG[1] 101: PWM2_OUT 110: I2S0_MCLK 111: BT_STATUS	0x0

GPIO_06_SEL

GPIO_06_SEL is the multiplex control register of the GPIO_06 pin.

Offset Address: 0x61C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_06_sel	GPIO_06 pin multiplex control 000: GPIO[6] 001: JTAG_MODE 010: UART1_TXD 011: SPI0_CK 100: DIAG[2] 101: PWM3_OUT 110: I2S0_TX 111: COEX_SWITCH	0x0

GPIO_07_SEL

GPIO_07_SEL is the multiplex control register of the GPIO_07 pin.

Offset Address: 0x620 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_07_sel	GPIO_07 pin multiplex control 000: GPIO[7] 001: HW_ID[5] 010: UART1_CTS_N 011: SPI0_RXD 100: DIAG[3] 101: PWM0_OUT 110: I2S0_BCLK 111: BT_ACTIVE	0x0

GPIO_08_SEL

GPIO_08_SEL is the multiplex control register of the GPIO_08 pin.

Offset Address: 0x Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_08_sel	GPIO_08 pin multiplex control 000: GPIO[8] 001: JTAG_ENABLE 010: UART1_RTS_N 011: SPI0_TXD 100: DIAG[4] 101: PWM1_OUT 110: I2S0_WS 111: WLAN_ACTIVE	0x0

GPIO_09_SEL

GPIO_09_SEL is the multiplex control register of the GPIO_09 pin.

Offset Address: 0x628 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_09_sel	GPIO_09 pin multiplex control 000: GPIO[9] 001: I2C0_SCL 010: UART2_RTS_N 011: SDIO_D2 100: SPI0_TXD 101: PWM0_OUT 110: DIAG[5] 111: I2S0_MCLK	0x0

GPIO_10_SEL

GPIO_10_SEL is the multiplex control register of the GPIO_10 pin.

Offset Address: 0x62C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_10_sel	GPIO_10 pin multiplex control 000: GPIO[10] 001: I2C0_SDA 010: UART2_CTS_N 011: SDIO_D3 100: SPI0_CK 101: PWM1_OUT 110: DIAG[6] 111: I2S0_TX	0x0

GPIO_11_SEL

GPIO_11_SEL is the multiplex control register of the GPIO_11 pin.

Offset Address: 0x630 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_11_sel	GPIO_11 pin multiplex control 000: GPIO[11] 001: HW_ID[6] 010: UART2_TXD 011: SDIO_CMD 100: SPI0_RXD 101: PWM2_OUT 110: RF_TX_EN_EXT 111: I2S0_RX	0x0

GPIO_12_SEL

GPIO_12_SEL is the multiplex control register of the GPIO_12 pin.

Offset Address: 0x634 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_12_sel	GPIO_12 pin multiplex control 000: GPIO[12] 001: HW_ID[7] 010: UART2_RXD 011: SDIO_CLK 100: SPI0_CSN 101: PWM3_OUT 110: RF_RX_EN_EXT 111: I2S0_BCLK	0x0

GPIO_13_SEL

GPIO_13_SEL is the multiplex control register of the GPIO_13 pin.

Offset Address: 0x638 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_13_sel	GPIO_13 pin multiplex control 000: SSI_DATA 001: UART0_TXD 010: UART2_RTS_N 011: SDIO_D0 100: GPIO[13] 101: PWM4_OUT 110: I2C0_SDA 111: I2S0_WS	0x0

GPIO_14_SEL

GPIO_14_SEL is the multiplex control register of the GPIO_14 pin.

Offset Address: 0x63C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	gpio_14_sel	GPIO_14 pin multiplex control 000: SSI_CLK 001: UART0_RXD 010: UART2_CTS_N 011: SDIO_D1 100: GPIO[14] 101: PWM5_OUT 110: I2C0_SCL 111: HW_ID[2]	0x0

SFC_CSN_SEL

SFC_CSN_SEL is the multiplex control register of the SFC_CSN pin.

Offset Address: 0x640 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	sfc_csn_sel	SFC_CSN pin multiplex control 000: SFC_CSN 001: SDIO_D2 010: GPIO[9] 011: DIAG[5] 100: SPI0_TXD Other values: reserved	0x0

SFC_IO1_SEL

SFC_IO1_SEL is the multiplex control register of the SFC_IO1 pin.

Offset Address: 0x644 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	sfc_io1_sel	SFC_I01 pin multiplex control 000: SFC_DO 001: SDIO_D3 010: GPIO[10] 011: DIAG[6] 100: SPI0_CK Other values: reserved	0x0

SFC_IO2_SEL

SFC_IO2_SEL is the multiplex control register of the SFC_IO2 pin.

Offset Address: 0x648 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	sfc_io2_sel	SFC_I02 pin multiplex control 000: SFC_WPN 001: SDIO_CMD 010: GPIO[11] 011: RF_TX_EN_EXT 100: SPI0_RXD Other values: reserved	0x0

SFC_IO0_SEL

SFC_IO0_SEL is the multiplex control register of the SFC_IO0 pin.

Offset Address: 0x64C Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	sfc_io0_sel	SFC_I00 pin multiplex control 000: SFC_DI 001: SDIO_CLK 010: GPIO[12] 011: RF_RX_EN_EXT 100: SPI0_CSN Other values: reserved	0x0

SFC_CLK_SEL

SFC_CLK_SEL is the multiplex control register of the SFC_CLK pin.

Offset Address: 0x650 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	sfc_clk_sel	SFC_CLK pin multiplex control 000: SFC_CLK 001: SDIO_D0 010: GPIO[13] 100: SSI_DATA Other values: reserved	0x0

SFC_IO3_SEL

SFC_IO3_SEL is the multiplex control register of the SFC_IO3 pin.

Offset Address: 0x654 Total Reset Value: 0x0000_0000

Bits	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2:0]	RW	sfc_io3_sel	SFC_IO3 pin multiplex control 000: SFC_HOLDN 001: SDIO_D1 010: GPIO[14] 100: SSI_CLK Other values: reserved	0x0

6.1.5 Summary of Control Registers

You can change the drive current strength of chip pins by configuring DS0, DS1, and DS2 (DS refers to drive strength). Table 6-6 and Table 6-7 show the mapping between the drive strength and DS0, DS1, and DS2. Names of control registers correspond to the names of chip pins.

Table 6-6 Drive strength of GPIO_00 to GPIO_11, GPIO_13, and GPIO_14

DS1	DS0	Drive Strength (mA)
0	0	8
0	1	6
1	0	4
1	1	2

Table 6-7 Drive strength of GPIO_12 and SFC pins

DS2	DS1	DS0	Drive Strength (mA)
0	0	0	16
0	0	1	14
0	1	0	12
0	1	1	10
1	0	0	8
1	0	1	6

DS2	DS1	DS0	Drive Strength (mA)
1	1	0	4
1	1	1	2

Table 6-8 lists the control registers.

Table 6-8 Summary of control registers (base address: 0x5000_A000)

Offset Address	Register	Description	Page
0x904	pad_gpio_00_ctrl	GPIO_00 functional pin control register	6-22
0x908	pad_gpio_01_ctrl	GPIO_01 functional pin control register	6-23
0x90C	pad_gpio_02_ctrl	GPIO_02 functional pin control register	6-24
0x910	pad_gpio_03_ctrl	GPIO_03 functional pin control register	6-25
0x914	pad_gpio_04_ctrl	GPIO_04 functional pin control register	6-26
0x918	pad_gpio_05_ctrl	GPIO_05 functional pin control register	6-27
0x91C	pad_gpio_06_ctrl	GPIO_06 functional pin control register	6-28
0x920	pad_gpio_07_ctrl	GPIO_07 functional pin control register	6-29
0x924	pad_gpio_08_ctrl	GPIO_08 functional pin control register	6-30
0x928	pad_gpio_09_ctrl	GPIO_09 functional pin control register	6-31
0x92C	pad_gpio_10_ctrl	GPIO_10 functional pin control register	6-32
0x930	pad_gpio_11_ctrl	GPIO_11 functional pin control register	6-33
0x934	pad_gpio_12_ctrl	GPIO_12 functional pin control register	6-34
0x938	pad_gpio_13_ctrl	GPIO_13 functional pin control register	6-35
0x93C	pad_gpio_14_ctrl	GPIO_14 functional pin control register	6-36
0x940	pad_sfc_csn_ctrl	SFC_CSN functional pin control register	6-37
0x944	pad_sfc_io1_ctrl	SFC_IO1 functional pin control register	6-38
0x948	pad_sfc_io2_ctrl	SFC_IO2 functional pin control register	6-39
0x94C	pad_sfc_io0_ctrl	SFC_IO0 functional pin control register	6-40
0x950	pad_sfc_clk_ctrl	SFC_CLK functional pin control register	6-41
0x954	pad_sfc_io3_ctrl	SFC_IO3 functional pin control register	6-42

6.1.6 Description of Control Registers

pad_gpio_00_ctrl

pad_gpio_00_ctrl is the functional pin control register of GPIO_00.

Offset Address: 0x904 Total Reset Value: 0x0E0B_44B0

Bits	Access	Name	Description	Reset
[31:28]	RW	reseverd	Reserved	0x0
[27]	RW	pad_gpio_00_ctrl_osc_ds2	Drive current control in quartz crystal unit (Xtal) mode [ds2,ds1,ds0]: 000: 0.6 μA 001: 0.7 μA 010: 0.8 μA 011: 0.9 μA 100: 1.0 μA 101: 1.2 μA 110: 1.4 μA 111: 1.6 μA	0x1
[26]	RW	pad_gpio_00_ctrl_osc_ds1		0x1
[25]	RW	pad_gpio_00_ctrl_osc_ds0		0x1
[24]	RW	reseverd	Reserved	0x0
[23]	RW	pad_gpio_00_ctrl_osc_ctl0	OSC control signal 0: external crystal oscillator 1: external crystal	0x0
[22]	RW	pad_gpio_00_ctrl_osc_e	RTC enable 0: disabled 1: enabled	0x0
[21:11]	RW	reserved	Reserved	0x0
[10]	RW	pad_gpio_00_ctrl_ie1	GPIO_00 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_00_ctrl_pd1	GPIO_00 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_00_ctrl_pu1	GPIO_00 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_00_ctrl_sr1	GPIO_00 slew rate 0: fast edge	0x1

Bits	Access	Name	Description	Reset
			1: slow edge	
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_00_ctrl_ds11	GPIO_00 driver control DS1	0x1
[4]	RW	pad_gpio_00_ctrl_ds01	GPIO_00 driver control DS0	0x1
[3]	RW	pad_gpio_00_ctrl_se1	GPIO_00 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_01_ctrl

pad_gpio_01_ctrl is the GPIO_01 functional pin control register.

Offset Address: 0x908 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_01_ctrl_ie	GPIO_01 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_01_ctrl_pd	GPIO_01 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_01_ctrl_pu	GPIO_01 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_01_ctrl_sr	GPIO_01 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_01_ctrl_ds1	GPIO_01 driver control DS1	0x1
[4]	RW	pad_gpio_01_ctrl_ds0	GPIO_01 driver control DS0	0x1
[3]	RW	pad_gpio_01_ctrl_se	GPIO_01 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0

Bits	Access	Name	Description	Reset
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_02_ctrl

pad_gpio_02_ctrl functional pin control register

Offset Address: 0x90C Total Reset Value: 0x0000_06B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_02_ctrl_ie	GPIO_02 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_02_ctrl_pd	GPIO_02 input pull-down enable 0: disabled 1: enabled	0x1
[8]	RW	pad_gpio_02_ctrl_pu	GPIO_02 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_02_ctrl_sr	GPIO_02 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_02_ctrl_ds1	GPIO_02 driver control DS1	0x1
[4]	RW	pad_gpio_02_ctrl_ds0	GPIO_02 driver control DS0	0x1
[3]	RW	pad_gpio_02_ctrl_se	GPIO_02 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_03_ctrl

pad_gpio_03_ctrl functional pin control register

Offset Address: 0x910 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_03_ctrl_ie	GPIO_03 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_03_ctrl_pd	GPIO_03 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_03_ctrl_pu	GPIO_03 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_03_ctrl_sr	GPIO_03 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_03_ctrl_ds1	GPIO_03 driver control DS1	0x1
[4]	RW	pad_gpio_03_ctrl_ds0	GPIO_03 driver control DS0	0x1
[3]	RW	pad_gpio_03_ctrl_se	GPIO_03 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_04_ctrl

pad_gpio_04_ctrl functional pin control register

Offset Address: 0x914 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_04_ctrl_ie	GPIO_04 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_04_ctrl_pd	GPIO_04 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_04_ctrl_pu	GPIO_04 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_04_ctrl_sr	GPIO_04 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_04_ctrl_ds1	GPIO_04 driver control DS1	0x1
[4]	RW	pad_gpio_04_ctrl_ds0	GPIO_04 driver control DS0	0x1
[3]	RW	pad_gpio_04_ctrl_se	GPIO_04 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_05_ctrl

pad_gpio_05_ctrl functional pin control register

Offset Address: 0x918 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_05_ctrl_ie	GPIO_05 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_05_ctrl_pd	GPIO_05 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_05_ctrl_pu	GPIO_05 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_05_ctrl_sr	GPIO_05 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_05_ctrl_ds1	GPIO_05 driver control DS1	0x1
[4]	RW	pad_gpio_05_ctrl_ds0	GPIO_05 driver control DS0	0x1
[3]	RW	pad_gpio_05_ctrl_se	GPIO_05 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_06_ctrl

pad_gpio_06_ctrl functional pin control register

Offset Address: 0x91C Total Reset Value: 0x0000_06B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_06_ctrl_ie	GPIO_06 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_06_ctrl_pd	GPIO_06 input pull-down enable 0: disabled 1: enabled	0x1
[8]	RW	pad_gpio_06_ctrl_pu	GPIO_06 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_06_ctrl_sr	GPIO_06 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_06_ctrl_ds1	GPIO_06 driver control DS1	0x1
[4]	RW	pad_gpio_06_ctrl_ds0	GPIO_06 driver control DS0	0x1
[3]	RW	pad_gpio_06_ctrl_se	GPIO_06 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_07_ctrl

pad_gpio_07_ctrl functional pin control register

Offset Address: 0x920 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_07_ctrl_ie	GPIO_07 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_07_ctrl_pd	GPIO_07 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_07_ctrl_pu	GPIO_07 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_07_ctrl_sr	GPIO_07 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_07_ctrl_ds1	GPIO_07 driver control DS1	0x1
[4]	RW	pad_gpio_07_ctrl_ds0	GPIO_07 driver control DS0	0x1
[3]	RW	pad_gpio_07_ctrl_se	GPIO_07 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_08_ctrl

pad_gpio_08_ctrl functional pin control register

Offset Address: 0x924 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_08_ctrl_ie	GPIO_08 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_08_ctrl_pd	GPIO_08 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_08_ctrl_pu	GPIO_08 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_08_ctrl_sr	GPIO_08 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_08_ctrl_ds1	GPIO_08 driver control DS1	0x1
[4]	RW	pad_gpio_08_ctrl_ds0	GPIO_08 driver control DS0	0x1
[3]	RW	pad_gpio_08_ctrl_se	GPIO_08 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_09_ctrl

pad_gpio_09_ctrlSTB_GPIO18_3 functional pin control register

Offset Address: 0x928 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_09_ctrl_ie	GPIO_09 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_09_ctrl_pd	GPIO_09 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_09_ctrl_pu	GPIO_09 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_09_ctrl_sr	GPIO_09 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_09_ctrl_ds1	GPIO_09 driver control DS1	0x1
[4]	RW	pad_gpio_09_ctrl_ds0	GPIO_09 driver control DS0	0x1
[3]	RW	pad_gpio_09_ctrl_se	GPIO_09 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_10_ctrl

pad_gpio_10_ctrl functional pin control register

Offset Address: 0x92C Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_10_ctrl_ie	GPIO_10 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_10_ctrl_pd	GPIO_10 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_10_ctrl_pu	GPIO_10 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_10_ctrl_sr	GPIO_10 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_10_ctrl_ds1	GPIO_10 driver control DS1	0x1
[4]	RW	pad_gpio_10_ctrl_ds0	GPIO_10 driver control DS0	0x1
[3]	RW	pad_gpio_10_ctrl_se	GPIO_10 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_11_ctrl

pad_gpio_11_ctrl functional pin control register

Offset Address: 0x930 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_11_ctrl_ie	GPIO_11 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_11_ctrl_pd	GPIO_11 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_11_ctrl_pu	GPIO_11 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_11_ctrl_sr	GPIO_11 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_11_ctrl_ds1	GPIO_11 driver control DS1	0x1
[4]	RW	pad_gpio_11_ctrl_ds0	GPIO_11 driver control DS0	0x1
[3]	RW	pad_gpio_11_ctrl_se	GPIO_11 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_12_ctrl

pad_gpio_12_ctrl functional pin control register

Offset Address: 0x934 Total Reset Value: 0x0000_04F0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_12_ctrl_ie	GPIO_12 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_12_ctrl_pd	GPIO_12 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_12_ctrl_pu	GPIO_12 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_12_ctrl_sr	GPIO_12 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	pad_gpio_12_ctrl_ds2	GPIO_12 driver control DS2	0x1
[5]	RW	pad_gpio_12_ctrl_ds1	GPIO_12 driver control DS1	0x1
[4]	RW	pad_gpio_12_ctrl_ds0	GPIO_12 driver control DS0	0x1
[3]	RW	pad_gpio_12_ctrl_se	GPIO_12 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_13_ctrl

pad_gpio_13_ctrl functional pin control register

Offset Address: 0x938 Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_13_ctrl_ie	GPIO_13 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_13_ctrl_pd	GPIO_13 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_13_ctrl_pu	GPIO_13 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_13_ctrl_sr	GPIO_13 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_13_ctrl_ds1	GPIO_13 driver control DS1	0x1
[4]	RW	pad_gpio_13_ctrl_ds0	GPIO_13 driver control DS0	0x1
[3]	RW	pad_gpio_13_ctrl_se	GPIO_13 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_gpio_14_ctrl

pad_gpio_14_ctrl functional pin control register

Offset Address: 0x93C Total Reset Value: 0x0000_04B0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_gpio_14_ctrl_ie	GPIO_14 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_gpio_14_ctrl_pd	GPIO_14 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_gpio_14_ctrl_pu	GPIO_14 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_gpio_14_ctrl_sr	GPIO_14 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	reserved	Reserved	0x0
[5]	RW	pad_gpio_14_ctrl_ds1	GPIO_14 driver control DS1	0x1
[4]	RW	pad_gpio_14_ctrl_ds0	GPIO_14 driver control DS0	0x1
[3]	RW	pad_gpio_14_ctrl_se	GPIO_14 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_sfc_csn_ctrl

pad_sfc_csn_ctrl functional pin control register

Offset Address: 0x940 Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_sfc_csn_ctrl_ie	SFC_CSN input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_sfc_csn_ctrl_pd	SFC_CSN input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_sfc_csn_ctrl_pu	SFC_CSN input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_sfc_csn_ctrl_sr	SFC_CSN slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	pad_sfc_csn_ctrl_ds2	SFC_CSN driver control DS2	0x1
[5]	RW	pad_sfc_csn_ctrl_ds1	SFC_CSN driver control DS1	0x1
[4]	RW	pad_sfc_csn_ctrl_ds0	SFC_CSN driver control DS0	0x0
[3]	RW	pad_sfc_csn_ctrl_se	SFC_CSN Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_sfc_io1_ctrl

pad_sfc_io1_ctrl functional pin control register

Offset Address: 0x944 Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_sfc_io1_ctrl_ie	SFC_IO1 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_sfc_io1_ctrl_pd	SFC_IO1 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_sfc_io1_ctrl_pu	SFC_IO1 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_sfc_io1_ctrl_sr	SFC_IO1 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	pad_sfc_io1_ctrl_ds2	SFC_IO1 driver control DS2	0x1
[5]	RW	pad_sfc_io1_ctrl_ds1	SFC_IO1 driver control DS1	0x1
[4]	RW	pad_sfc_io1_ctrl_ds0	SFC_IO1 driver control DS0	0x0
[3]	RW	pad_sfc_io1_ctrl_se	SFC_IO1 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_sfc_io2_ctrl

pad_sfc_io2_ctrl functional pin control register

Offset Address: 0x948 Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_sfc_io2_ctrl_ie	SFC_IO2 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_sfc_io2_ctrl_pd	SFC_IO2 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_sfc_io2_ctrl_pu	SFC_IO2 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_sfc_io2_ctrl_sr	SFC_IO2 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	pad_sfc_io2_ctrl_ds2	SFC_IO2 driver control DS2	0x1
[5]	RW	pad_sfc_io2_ctrl_ds1	SFC_IO2 driver control DS1	0x1
[4]	RW	pad_sfc_io2_ctrl_ds0	SFC_IO2 driver control DS0	0x0
[3]	RW	pad_sfc_io2_ctrl_se	SFC_IO2 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_sfc_io0_ctrl

pad_sfc_io0_ctrl functional pin control register

Offset Address: 0x94C Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_sfc_io0_ctrl_ie	SFC_IO0 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_sfc_io0_ctrl_pd	SFC_IO0 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_sfc_io0_ctrl_pu	SFC_IO0 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_sfc_io0_ctrl_sr	SFC_IO0 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	pad_sfc_io0_ctrl_ds2	SFC_IO0 driver control DS2	0x1
[5]	RW	pad_sfc_io0_ctrl_ds1	SFC_IO0 driver control DS1	0x1
[4]	RW	pad_sfc_io0_ctrl_ds0	SFC_IO0 driver control DS0	0x0
[3]	RW	pad_sfc_io0_ctrl_se	SFC_IO0 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_sfc_clk_ctrl

pad_sfc_clk_ctrl functional pin control register

Offset Address: 0x950 Total Reset Value: 0x0000_04D0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_sfc_clk_ctrl_ie	SFC_CLK input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_sfc_clk_ctrl_pd	SFC_CLK input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_sfc_clk_ctrl_pu	SFC_CLK input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_sfc_clk_ctrl_sr	SFC_CLK slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	pad_sfc_clk_ctrl_ds2	SFC_CLK driver control DS2	0x1
[5]	RW	pad_sfc_clk_ctrl_ds1	SFC_CLK driver control DS1	0x0
[4]	RW	pad_sfc_clk_ctrl_ds0	SFC_CLK driver control DS0	0x1
[3]	RW	pad_sfc_clk_ctrl_se	SFC_CLK Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

pad_sfc_io3_ctrl

pad_sfc_io3_ctrl functional pin control register

Offset Address: 0x954 Total Reset Value: 0x0000_04E0

Bits	Access	Name	Description	Reset
[31:11]	RW	reserved	Reserved	0x000000
[10]	RW	pad_sfc_io3_ctrl_ie	SFC_IO3 input signal enable 0: disabled 1: enabled	0x1
[9]	RW	pad_sfc_io3_ctrl_pd	SFC_IO3 input pull-down enable 0: disabled 1: enabled	0x0
[8]	RW	pad_sfc_io3_ctrl_pu	SFC_IO3 input pull-up enable 0: disabled 1: enabled	0x0
[7]	RW	pad_sfc_io3_ctrl_sr	SFC_IO3 slew rate 0: fast edge 1: slow edge	0x1
[6]	RW	pad_sfc_io3_ctrl_ds2	SFC_IO3 driver control DS2	0x1
[5]	RW	pad_sfc_io3_ctrl_ds1	SFC_IO3 driver control DS1	0x1
[4]	RW	pad_sfc_io3_ctrl_ds0	SFC_IO3 driver control DS0	0x0
[3]	RW	pad_sfc_io3_ctrl_se	SFC_IO3 Schmitt trigger enable 0: no Schmitt 1: Schmitt enable	0x0
[2:0]	RW	reserved	Reserved	0x0

6.2 GPIO

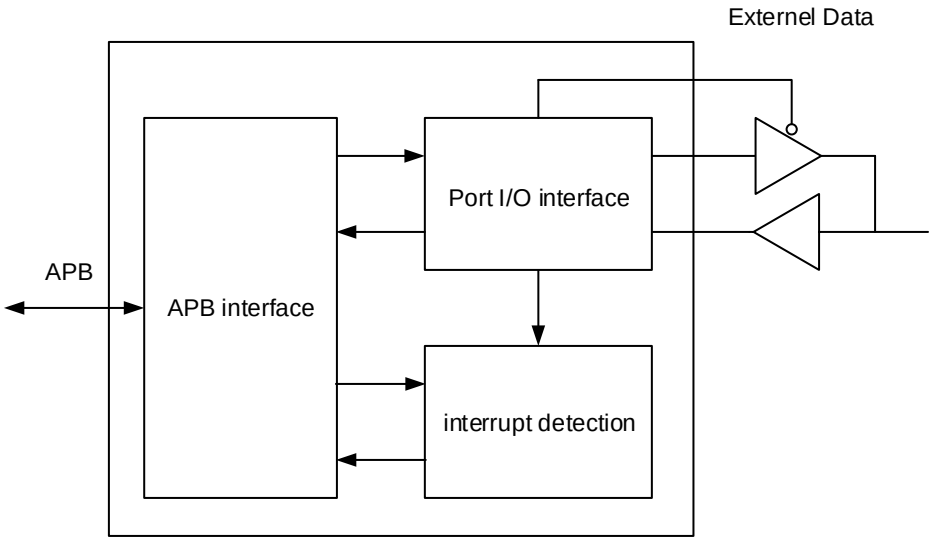
6.2.1 Overview

The general purpose input/output (GPIO) generates output signals for or collects input signals from specific applications to enable communication between the system and peripherals so that the system can control peripherals. GPIOs of Hi3861, Hi3861L, and Hi3881 comply with the APB protocol of AMBA2.0.

As shown in Figure 6-1, a GPIO module has the following interfaces:

- APB interface
- I/O pad interface for external data
- Interrupt detection interface

Figure 6-1 GPIO schematic diagram



6.2.2 Features

The GPIO interface has the following features:

- The clock source can be the 24 MHz/40 MHz crystal clock in working mode or 32K clock in low-power mode.
- One group of GPIO consists of 15 independent configurable pins.
- Each GPIO pin separately controls the transfer direction.
- Each GPIO pin can be separately configured as an external interrupt source.
- When the GPIO is used as an interrupt, there are four interrupt triggering modes as follows:
  - Triggered by rising edge
  - Triggered by falling edge
  - Triggered by high level

  - Triggered by low level
- When the GPIO reports an interrupt, and the CPU queries the GPIO ID.
- Each interrupt can be separately masked, and pulse interrupts can be cleared.

6.2.3 Operating Modes

The GPIO can be configured as the input or output by configuring GPIO_SWPORT_DDR.

- If the GPIO is configured as the input, it can be input to the input port register GPIO_EXT_PORT, and can be used as an external interrupt source or external sleep wakeup signal.
- If the GPIO is configured as the output, configuration data can be input to pins by configuring GPIO_SWPORT_DR.

The interrupt triggering mode can be set to edge triggering or level triggering by configuring GPIO_INTTYPE_LEVEL.

Level triggering can be set to high-level triggering or low-level triggering by configuring GPIO_INT_PLOARITY.

Single-edge triggering can be set to rising-edge triggering or falling-edge triggering by configuring GPIO_INT_PLOARITY.

Each interrupt can be separately enabled by configuring GPIO_INTEN.

Each interrupt can be separately masked by configuring GPIO_INTMASK.

The status of each interrupt can be queried.

- Query the raw interrupt status before masking, by calling GPIO_RAWINTSTATUS.
- Query the final interrupt status after masking by calling GPIO_INTSTATUS.

The pulse interrupt can be cleared by configuring GPIO_PORT_EOI.

NOTICE

To prevent the generation of unnecessary interrupts and wakeup signals, configure the GPIO as an interrupt source by bit. You only need to configure the GPIO pins that are required to function as interrupts.

6.2.4 Register Summary

Table 6-9 describes the GPIO registers.

Table 6-9 Summary of the GPIO registers (base address: 0x5000_6000)

Offset Address	Register	Description	Page
0x00	GPIO_SWPORT_DR	Port data output register	6-45
0x04	GPIO_SWPORT_DDR	Port data transfer direction register	6-45
0x30	GPIO_INTEN	Port interrupt enable register	6-46

Offset Address	Register	Description	Page
0x34	GPIO_INTMASK	Port interrupt mask register	6-46
0x38	GPIO_INTTYPE_LEVEL	Port interrupt type register	6-46
0x3C	GPIO_INT_PLOARITY	Port interrupt polarity register	6-47
0x40	GPIO_INTSTATUS	Port interrupt status register	6-47
0x44	GPIO_RAWINTSTATUS	Port raw interrupt status register	6-47
0x4C	GPIO_PORT_EOI	Port interrupt clear register	6-48
0x50	GPIO_EXT_PORT	Port data input interface register	6-48

6.2.5 Register Description

GPIO_SWPORT_DR

GPIO_SWPORT_DR is the port data output register.

Offset Address: 0x00 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_swport_dr	Data output. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively.	0x0000

GPIO_SWPORT_DDR

GPIO_SWPORT_DDR is the port data transfer direction register.

Offset Address: 0x04 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_swport_ddr	Data transfer direction, which determines whether a bit is input or output. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively. The meaning of each bit is as follows: 0: input (default value) 1: output	0x0000

GPIO_INTEN

GPIO_INTEN is the port interrupt enable register.

Offset Address: 0x30 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_inten	Interrupt enable, which is used to determine whether a bit of the port signal is in interrupt mode or normal mode. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively. The meaning of each bit is as follows: 0: normal mode (default value) 1: interrupt mode	0x0000

GPIO_INTMASK

GPIO_INTMASK is a port interrupt mask register.

Offset Address: 0x34 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_intmask	Interrupt mask, which determines whether a bit of the corresponding port is masked by an interrupt. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively. The meaning of each bit is as follows: 0: not masked (default value) 1: masked	0x0000

GPIO_INTTYPE_LEVEL

GPIO_INTTYPE_LEVEL is the port interrupt type register.

Offset Address: 0x38 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_inttype_level	Interrupt triggering type, which determines the interrupt type. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively. The meaning of each bit is as follows: 0: level-triggered (default value) 1: edge-triggered	0x0000

GPIO_INT_PLOARITY

GPIO_INT_PLOARITY is the port interrupt polarity register.

Offset Address: 0x3C Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RW	gpio_int_ploarity	Interrupt polarity. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively. The meaning of each bit is as follows: 0: falling edge- or low level-triggered (default value) 1: rising edge- or high level-triggered	0x0000

GPIO_INTSTATUS

GPIO_INTSTATUS is the port interrupt status register. If the GPIO is configured to be interrupt-triggered, this register indicates the interrupt status of GPIO input signals.

Offset Address: 0x40 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RO	gpio_intstatus	Interrupt status, indicating whether the interrupt of a bit is generated. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively. The meaning of each bit is as follows: 0: No interrupt is generated. 1: The interrupt is generated.	0x0000

GPIO_RAWINTSTATUS

GPIO_RAWINTSTATU is the port raw interrupt status register. If the GPIO is configured to be interrupt-triggered, this register indicates the raw interrupt status of GPIO input signals.

Offset Address: 0x44 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RO	gpio_rawintstatus	Raw interrupt status, indicating whether the interrupt of a bit is generated before interrupt masking. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively. The meaning of each bit is as follows: 0: No interrupt is generated.	0x0000

Bits	Access	Name	Description	Reset
			1: The interrupt is generated.	

GPIO_PORT_EOI

GPIO_PORT_EOI is the port interrupt clear register. This register clears GPIO input interrupts.

Offset Address: 0x4C Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	WO	gpio_port_eoi	Interrupt clear, indicating whether a port interrupt is cleared. Interrupts are not automatically cleared. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively. The meaning of each bit is as follows: 0: not cleared 1: cleared	0x0000

GPIO_EXT_PORT

GPIO_EXT_PORT is the port data input interface register.

Offset Address: 0x50 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:0]	RO	gpio_ext_port	Stores the input data of ports that are configured as inputs. bit[0] to bit[15] correspond to GPIO_0 to GPIO_15 respectively.	0x0000

6.3 UART

6.3.1 Overview

The universal asynchronous receiver/transmitter (UART) is an asynchronous serial communications port, which connects to the UART of an external chip to implement communication between the chips.

A chip provides three UART units and supports the 2-wire mode.

6.3.2 Features

The UART has the following features:

- Supports 64 x 8-bit TX first in first out (FIFO) and 64 x 12-bit RX FIFO.
- Supports programmable data bit width and stop bit width.
  - Sets the data bit width to 5, 6, 7, or 8 bits by programming.
  - Sets the stop bit width to 1 or 2 bits by programming.
- Supports parity check or no check.
- Supports programmable transfer rate and integral and decimal frequency division.
- Supports RX FIFO interrupts, TX FIFO interrupts, RX timeout interrupts, and error interrupts.
- Queries the status of raw or masked interrupts.
- Allows the UART or the RX/TX function of the UART to be disabled by programming to reduce power consumption.
- UART0 does not support hardware flow control. UART1 and UART2 support hardware flow control.

6.3.3 Operating Modes

Interface Signals

Table 6-10 lists the UART interface signals.

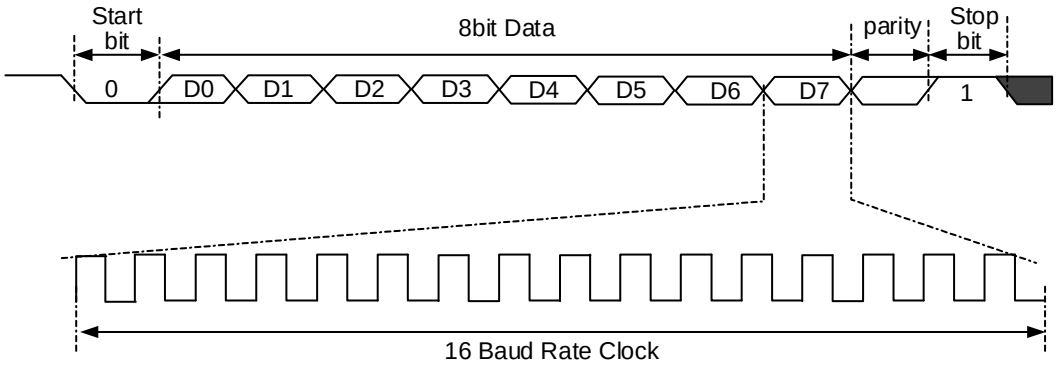
Table 6-10 UART interface signals

Signal	Width (Bit)	Direction	Description
RXD	1	I	Data input
TXD	1	O	Data output
CTS	1	I	Clear-to-send (CTS) signal, for hardware flow control and active low
RTS	1	O	Request-to-send (RTS) signal, for hardware flow control and active low

UART Data Frame Format

Figure 6-2 shows the UART data frame format. The data frame length, number of stop bits, and parity check are configurable.

Figure 6-2 Format of a UART data frame



UART Initialization Configuration

To initialize the UART, perform the following steps:

Step 1 Read UART_FR[busy], wait until the UART is idle, and perform a soft reset on the UART module.

Step 2 Set UART_CR[uarten] to **0** to disable the UART.

Step 3 Set UART_LCR_H to **0** to clear the UART configuration.

Step 4 Configure UART_IBRD and UART_FBRD based on the baud rate, and set the baud rate frequency divider value.

Step 5 Set the parity bit, data length, FIFO enable, and the number of stop bits by configuring UART_LCR_H as required.

Step 6 Configure UART_IFLS to set the TX FIFO threshold, RX FIFO threshold, and flow control threshold.

Step 7 If direct memory access (DMA) is required for data transfer, write **1** to **[txdmae]** or **[rxdmae]** to enable the DMA of the UART.

Step 8 To enable the flow control function of the UART hardware, write **1** to **[cts_en]** and **[rts_en]**.

Step 9 Set **[rxe]** to **1** to enable UART RX. Set **[txe]** to **1** to enable the UART TX. Set **[uarten]** to **1** to enable the UART.

----End

Data transfer starts after register initialization.

NOTICE

- The values of UART_IBRD and UART_FBRD are updated only after the current data transmission or reception is complete.
- The minimum frequency divider value is **1**, and the maximum frequency divider value is **65535 (0xFFFF)**. That is, UART_IBRD=**0** is invalid and UART_FBRD will be ignored. If UART_IBRD=**65535 (0xFFFF)**, the value of UART_FBRD must be **0**. Otherwise, the transmission and reception may fail.

NOTE

The baud rate frequency divider is calculated as follows:

- Calculate the theoretical frequency divider *n*. UART baud rate = Internal bus frequency/(16 x *n*).
- The integral part of the frequency divider (UART_IBRD) is set to **integer(n)**.
- The fractional part of the frequency divider is **integer((n-integer (n)) ×64+0.5)**, and **integer()** is a round-down function.

6.3.4 Register Summary

Table 6-11 describes UART registers.

Table 6-11 Summary of UART registers (UART0 base address: 0x4000_8000; UART1 base address: 0x4000_9000; UART2 base address: 0x4000_A000)

Offset Address	Register	Description	Page
0x000	UART_DR	UART data register	6-52
0x004	UART_RSR	RX state register or error clear register	6-52
0x018	UART_FR	UART flag register	6-53
0x024	UART_IBRD	Integer baud rate register	6-55
0x028	UART_FBRD	Fractional baud rate register	6-55
0x02C	UART_LCR_H	Transfer mode control register	6-56
0x030	UART_CR	UART control register	6-57
0x034	UART_IFLS	Interrupt FIFO threshold selection register	6-58
0x038	UART_IMSC	Interrupt mask register	6-59
0x03C	UART_RIS	Raw Interrupt Status Register	6-60
0x040	UART_MIS	Interrupt state register after mask	6-61
0x044	UART_ICR	Interrupt clear register	6-62
0x048	UART_DMACR	DMA control register	6-63

6.3.5 Register Description

UART_DR

UART_DR is the UART data register. It is used to store data to be received and transmitted. The RX status can be queried by reading this register.

Offset Address: 0x000 Total Reset Value: 0x0000

Bits	Access	Name	Description	Reset
[15:12]	-	reserved	Reserved	0x0
[11]	RO	oe	Overflow error status 0: No overflow error occurs. 1: An overflow error occurs. That is, a data entry is received when the RX FIFO is full.	0x0
[10]	RO	be	Break error status 0: No break error occurs. 1: A break error occurs. For the RX data input signal, if the period of remaining low is longer than that of transferring a full word (a start bit, a data bit, a parity bit, and a stop bit), a break error occurs.	0x0
[9]	RO	pe	Parity error status 0: No parity error occurs. 1: A parity error occurs.	0x0
[8]	RO	fe	Frame error status 0: No frame error is detected. 1: A frame error occurs.	0x0
[7:0]	RW	data	Data to be transmitted and received	0x00

UART_RSR

NOTICE

Any write operation on the UART_RSR resets the UART_RSR.

UART_RSR is the RX status register or error clear register. It is used as the RX status register when being read, and is used as the error clear register when being written.

You can obtain the RX state by reading UART_DR as well. The break, frame, and parity state information read from UART_DR takes priority over that read from UART_RSR. That is, the state information in UART_DR updates faster than that in UART_RSR.

Offset Address: 0x004 Total Reset Value: 0x00

Bit	Access	Name	Description	Reset
[7:4]	-	reserved	Reserved	0x0
[3]	RW	oe	Overflow error status and clear 0: No overflow error occurs. 1: An overflow error occurs. When the FIFO is full, the content 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.	0x0
[2]	RW	be	Break error status and clear 0: No break error occurs. 1: A break error occurs. For the RX data input signal, if the period of remaining low is longer than that of transferring a full word (a start bit, data bit, parity bit, and stop bit), a break error occurs.	0x0
[1]	RW	pe	Parity error status and clear 0: No parity error occurs. 1: A parity error of the received data occurs.	0x0
[0]	RW	fe	Frame error status and clear 0: No frame error is detected. 1: An error occurs at the stop bit of the received data. The valid stop bit is high level.	0x0

UART_FR

UART_FR is the UART flag register.

Offset Address: 0x018 Total Reset Value: 0x0197

Bit	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x01
[7]	RO	txfe	TX FIFO empty status When UART_LCR_H[fen] is set to **0**: 0: The TX holding register is not empty. 1: The TX holding register is empty. When UART_LCR_H[fen] is set to **1**: 0: The TX FIFO is not empty. 1: The TX FIFO is empty.	0x1

Bit	Access	Name	Description	Reset
[6]	RO	rxff	RX FIFO full status When UART_LCR_H[fen] is set to **0**: 0: The RX holding register is not full. 1: The RX holding register is full. When UART_LCR_H[fen] is set to **1**: 0: The RX FIFO is not full. 1: The RX FIFO is full.	0x0
[5]	RO	txff	TX FIFO full status When UART_LCR_H[fen] is set to **0**: 0: The TX holding register is not full. 1: The TX holding register is full. When UART_LCR_H[fen] is set to **1**: 0: The TX FIFO is not full. 1: The TX FIFO is not full.	0x0
[4]	RO	rxfe	RX FIFO empty status When UART_LCR_H[fen] is set to **0**: 0: The RX holding register is not empty. 1: RX holding register empty. When UART_LCR_H[fen] is set to **1**: 0: The RX FIFO is not empty. 1: The RX FIFO is empty.	0x1
[3]	RO	busy	UART busy/idle state 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 state is retained until the entire byte (including all stop bits) is transmitted from the shift register. When the TX FIFO is not empty, this bit is set to **1**.	0x0
[2:0]	-	reserved	Reserved	0x7

UART_IBRD

UART_IBRD is the integral baud rate register.

Offset Address: 0x024 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:0]	RW	bauddivint	Frequency divider value for the integral part of the baud rate. **Note: All bits are cleared after reset.**	0x0000

UART_FBRD

UART_FBRD is the fractional baud rate register.

The values of UART_IBRD and UART_FBRD are updated only after the current data transmission and reception are complete.

The minimum frequency divider value is **1**, and the maximum frequency divider value is **65535 (2^16 – 1)**. **UART_IBRD=0** is invalid and UART_FBRD will be ignored. If UART_IBRD**=65535 (0xFFFF)**, the value of UART_FBRD must be **0**. Otherwise, the transmission and reception fail. The baud rate frequency divider is calculated as follows:

Calculate the theoretical frequency divider *n*. UART baud rate = Internal bus frequency/(16 x *n*).

The integral part of the frequency divider (UART_IBRD) is set to **integer(n)**.

The fractional part of the frequency divider is **integer((n-integer (n)) x 64 + 0.5)**, and **integer()** is a round-down function.

Offset Address: 0x028 Total Reset Value: 0x00

Bit	Access	Name	Description	Reset
[7:6]	-	reserved	Reserved	0x0
[5:0]	RW	banddivfrac	Frequency divider value for the fractional part of the baud rate. **Note: All bits are cleared after reset.**	0x00

UART_LCR_H

UART_LCR_H is the transfer mode control register.

NOTE
If UART_IBRD and UART_FBRD are updated, UART_LCR_H must be updated.

Offset Address: 0x02C Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x00
[7]	RW	sps	Parity bit enable 0: Stick Parity disabled 1: When **[pen]** and **[eps]** are set to **1**, the parity bit is transmitted and checked as **0**. When **[pen]** is set to **1** and **[eps]** is set to **0**, the parity bit is transmitted and checked as **1**.	0x0
[6:5]	RW	wlen	Number of data bits in a frame to be sent or received 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits	0x0
[4]	RW	fen	TX/RX FIFO enable 0: disabled 1: enabled	0x0
[3]	RW	stp2	Whether a 2-bit stop bit exists at the end of a transmitted frame 0: No 2-bit stop bit exists. 1: A 2-bit stop bit exists. **Note: According to the RX logic, whether a 2-bit stop bit exists at the frame end is not checked during data reception.**	0x0
[2]	RW	eps	Parity bit select during data transmission and reception 0: odd parity generation and checking 1: even parity generation and checking **Note: When [pen] is 0, this bit is invalid.**	0x0
[1]	RW	pen	Parity check select 0: The parity check is disabled. 1: The parity bit is generated at the TX end and checked at the RX end.	0x0
[0]	RW	brk	TX break select 0: invalid	0x0

Bit	Access	Name	Description	Reset
			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 the UART control register.

Offset Address: 0x030 Total Reset Value: 0x0300

Bit	Access	Name	Description	Reset
[15]	RW	ctsen	CTS hardware flow control enable 0: disabled 1: enabled (Data is transmitted only when the nUARTCTS signal is valid.)	0x0
[14]	RW	rtsen	RTS hardware flow control enable 0: disabled 1: enabled (The data reception is requested only when the RX FIFO has free space.)	0x0
[13:12]	-	reserved	Reserved	0x0
[11]	RW	rts	Request transmission setting This bit is the inversion of the status output signal nUARTRTS of the UART modem. 0: The output signal remains unchanged. 1: The output signal is **0**.	0x0
[10]	RW	dtr	Data transmission preparation setting This bit is the inversion of the status output signal nUARTDTR of the UART modem. 0: The output signal remains unchanged. 1: The output signal is **0**.	0x0
[9]	RW	rxe	UART RX enable 0: disabled 1: enabled **Note: If the UART is disabled during data reception, the current data reception is stopped abnormally.**	0x1
[8]	RW	txe	UART TX enable	0x1

Bit	Access	Name	Description	Reset
			0: disabled 1: enabled **Note: If the UART is disabled during data transmission, the current data transmission is stopped abnormally.**	
[7]	RW	lbe	Loopback enable 0: disabled 1: The UARTTXD output is looped back to UARTRXD.	0x0
[6:1]	-	reserved	Reserved	0x00
[0]	RW	uarten	UART enable 0: disabled 1: enabled **Note: If the UART is disabled during data reception and transmission, the current data transmission is stopped abnormally.**	0x0

UART_IFLS

UART_IFLS is the interrupt FIFO threshold select register. It is used to set the trigger line of the FIFO interrupt (UART_TXINTR or UART_RXINTR).

Offset Address: 0x034 Total Reset Value: 0x0092

Bit	Access	Name	Description	Reset
[15:9]	-	reserved	Reserved	0x00
[8:6]	RW	rtsflsel	Hardware flow control **rts_n** trigger condition 000: RX FIFO ≥ 1/8 full 001: RX FIFO ≥ 1/4 full 010: RX FIFO ≥ 1/2 full 011: RX FIFO ≥ 3/4 full 100: RX FIFO ≥ 7/8 full 101–111: reserved	0x2
[5:3]	RW	rxiflsel	RX interrupt FIFO threshold. An RX interrupt is triggered when any of the following conditions is met: 000: RX FIFO ≥ 1/8 full 001: RX FIFO ≥ 1/4 full 010: RX FIFO ≥ 1/2 full 011: RX FIFO ≥ 3/4 full 100: RX FIFO ≥ 7/8 full	0x2

Bit	Access	Name	Description	Reset
			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 ≤ 1/8 full 001: TX FIFO ≤ 1/4 full 011: TX FIFO ≤ 3/4 full 010: TX FIFO ≤ 1/2 full 100: TX FIFO ≤ 7/8 full 101–111: reserved	0x2

UART_IMSC

INT_MASK is the interrupt mask register.

Offset Address: 0x038 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	RW	oeim	Overflow error interrupt mask 0: enabled 1: disabled	0x0
[9]	RW	beim	Break error interrupt mask 0: enabled 1: disabled	0x0
[8]	RW	peim	Check interrupt mask 0: enabled 1: disabled	0x0
[7]	RW	feim	Break error interrupt mask 0: enabled 1: disabled	0x0
[6]	RW	rtim	RX timeout interrupt mask 0: enabled 1: disabled	0x0
[5]	RW	txim	TX interrupt mask 0: enabled 1: disabled	0x0
[4]	RW	rxim	RX interrupt mask	0x0

Bit	Access	Name	Description	Reset
			0: enabled 1: disabled	
[3:0]	-	reserved	Reserved	0x0

UART_RIS

UART_RIS is the raw interrupt state register.

NOTE
Its value is not affected by the interrupt mask register.

Offset Address: 0x03C Total Reset Value: 0x000F

Bit	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	RO	oeris	Raw overflow error interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[9]	RO	beris	Raw break error interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	peris	Raw parity interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7]	RO	feris	Raw error interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	RO	rtris	Raw RX timeout interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[5]	RO	txris	Raw TX interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	rxris	Raw RX interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3:0]	-	reserved	Reserved	0xF

UART_MIS

UART_MIS is the masked interrupt status register.

NOTE
The content is the result obtained after the raw interrupt status is ANDed with the interrupt mask state.

Offset Address: 0x040 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	RO	oemis	Masked overflow error interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[9]	RO	bemis	Masked break error interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	pemis	Masked parity interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7]	RO	femis	Masked error interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[6]	RO	rtmis	Masked RX timeout interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[5]	RO	txmis	Masked TX interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	rxmis	Masked RX interrupt state 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3:0]	-	reserved	Reserved	0x0

UART_ICR

UART_ICR is the interrupt clear register.

NOTE
Writing **1** clears the corresponding interrupt, and writing **0** has no effect.

Offset Address: 0x044 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	WO	oeic	Overflow error interrupt clear 0: invalid 1: cleared	0x0
[9]	WO	beic	Break error interrupt clear 0: invalid 1: cleared	0x0
[8]	WO	peic	Parity interrupt clear 0: invalid 1: cleared	0x0
[7]	WO	feic	Error interrupt clear 0: invalid 1: cleared	0x0
[6]	WO	rtic	RX timeout interrupt clear 0: invalid 1: cleared	0x0
[5]	WO	txic	TX interrupt clear 0: invalid 1: cleared	0x0
[4]	WO	rxic	RX interrupt clear 0: invalid 1: cleared	0x0
[3:0]	-	reserved	Reserved	0x0

UART_DMACR

UART_DMACR is the DMA control register. It is used to control the DMA for the TX FIFO and RX FIFO.

Offset Address: 0x048 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:3]	-	reserved	Reserved	0x0000
[2]	RW	dmaonerr	DMA enable control for the RX channel when the UART error interrupt (UARTEINTR) occurs 0: The request output (UARTRXDMASREQ or UARRTXDMABREQ) is valid. 1: The request output (UARTRXDMASREQ or UARRTXDMABREQ) is invalid.	0x0
[1]	RW	txdmae	TX FIFO DMA enable 0: disabled 1: enabled	0x0
[0]	RW	rxdmae	RX FIFO DMA enable 0: disabled 1: enabled	0x0

6.4 I2C

6.4.1 Overview

The I^2C module works as a slave device on the APB bus while a master device on the I^2C bus. The I^2C module reads and writes data from and to the slave device on the I^2C bus. The CPU can continuously configure multiple TX data entries and receive multiple data entries. The I^2C bus can mount multiple slave devices.

6.4.2 Features

The I^2C module has the following features:

- Supports the I^2C bus protocol V2.0 and only the master mode.
- Acts as a slave device on the APB bus while a master device on the I^2C bus and supports bus arbitration when there are multiple master devices.
- The I^2C master can write data to the slave or receive data from the slave.
- Supports clock synchronization as well as bit and byte waiting.
- Supports the interrupt or polling operation.
- Supports standard addresses (7 bits) and extended addresses (10 bits).
- Works in two speed modes: standard mode (100 kbit/s) and fast mode (400 kbit/s).

- Supports general call and start byte.
- Does not support the CBUS component.
- Filters the received serial data and address (SDA) and serial clock line (SCL) signals.
- Contains one 32 x 8-bit TX FIFO and one 32 x 8-bit RX FIFO.
- Allows hardware to detect the FIFO depth and reports corresponding interrupts.
- Supports both the FIFO and non-FIFO working modes.

6.4.3 Operating Modes

The I^2C module works in the following scenarios:

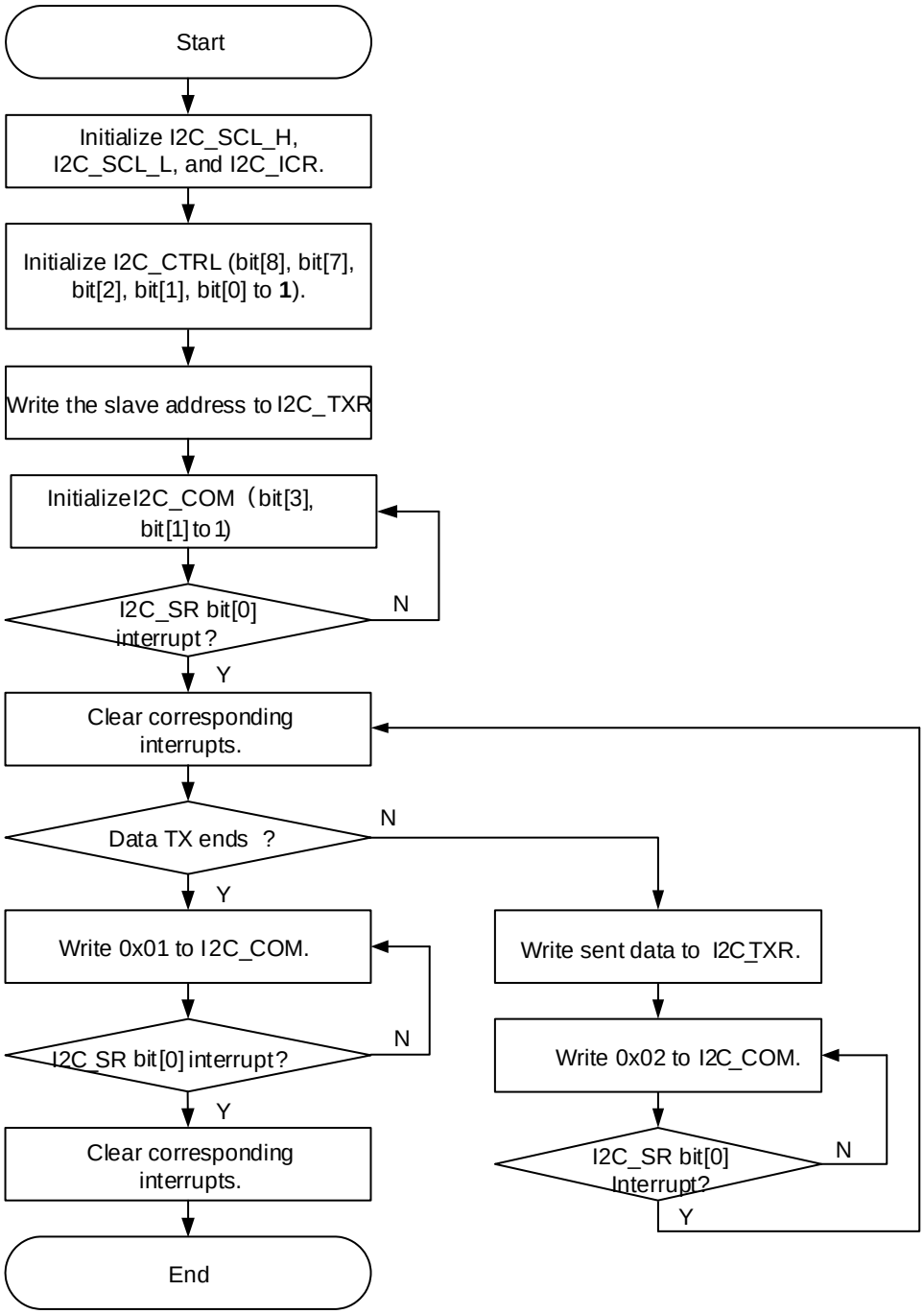
- The host transmits and receives only a single data entry (FIFO is not used).
- The host continuously transmits and receives multiple data entries (FIFO is used).

6.4.3.1 When FIFO Is Not Used

Data Transmission Process of the I^2C Master

Figure 6-3 shows the process for transmitting data by using the I^2C master.

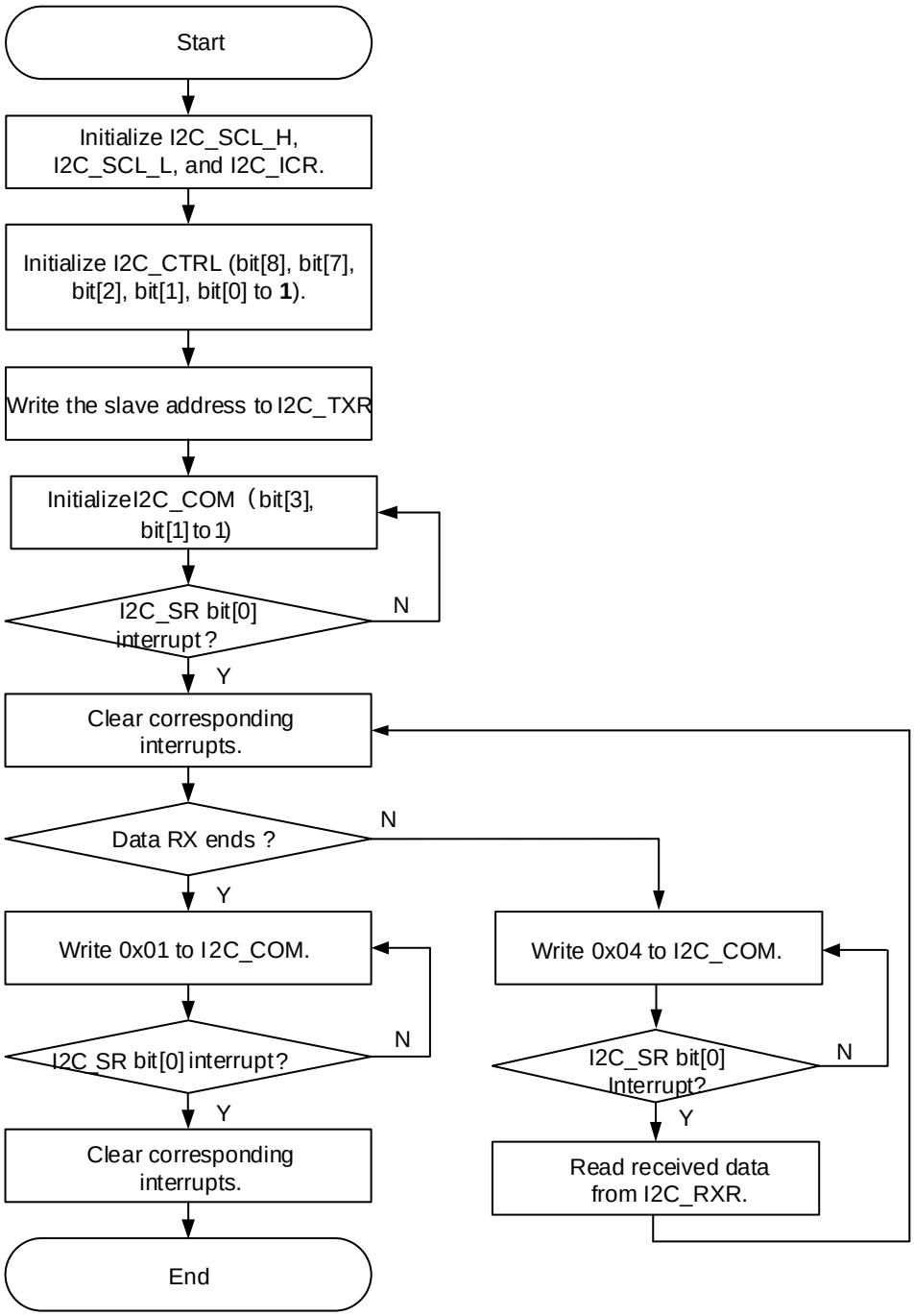
Figure 6-3 Data transmission (FIFO not used)



Data Reception Process of the I²C Master

Figure 6-4 shows the process of receiving data by using the I²C master.

Figure 6-4 Data reception (FIFO not used)

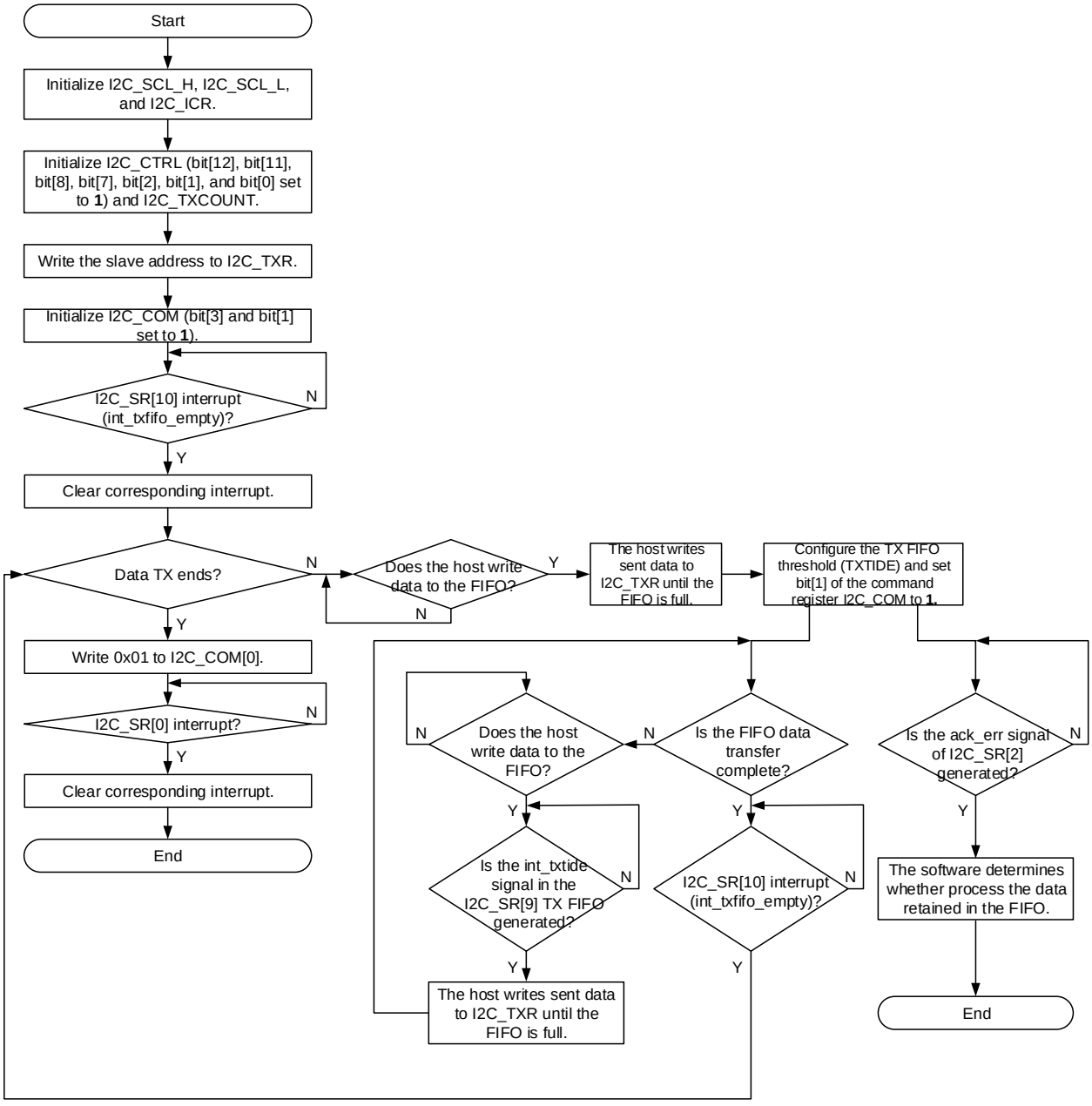


6.4.3.2 When FIFO Is Used

Data Transmission Process of the I²C Master

Figure 6-5 shows the process of transmitting data by using the I²C master.

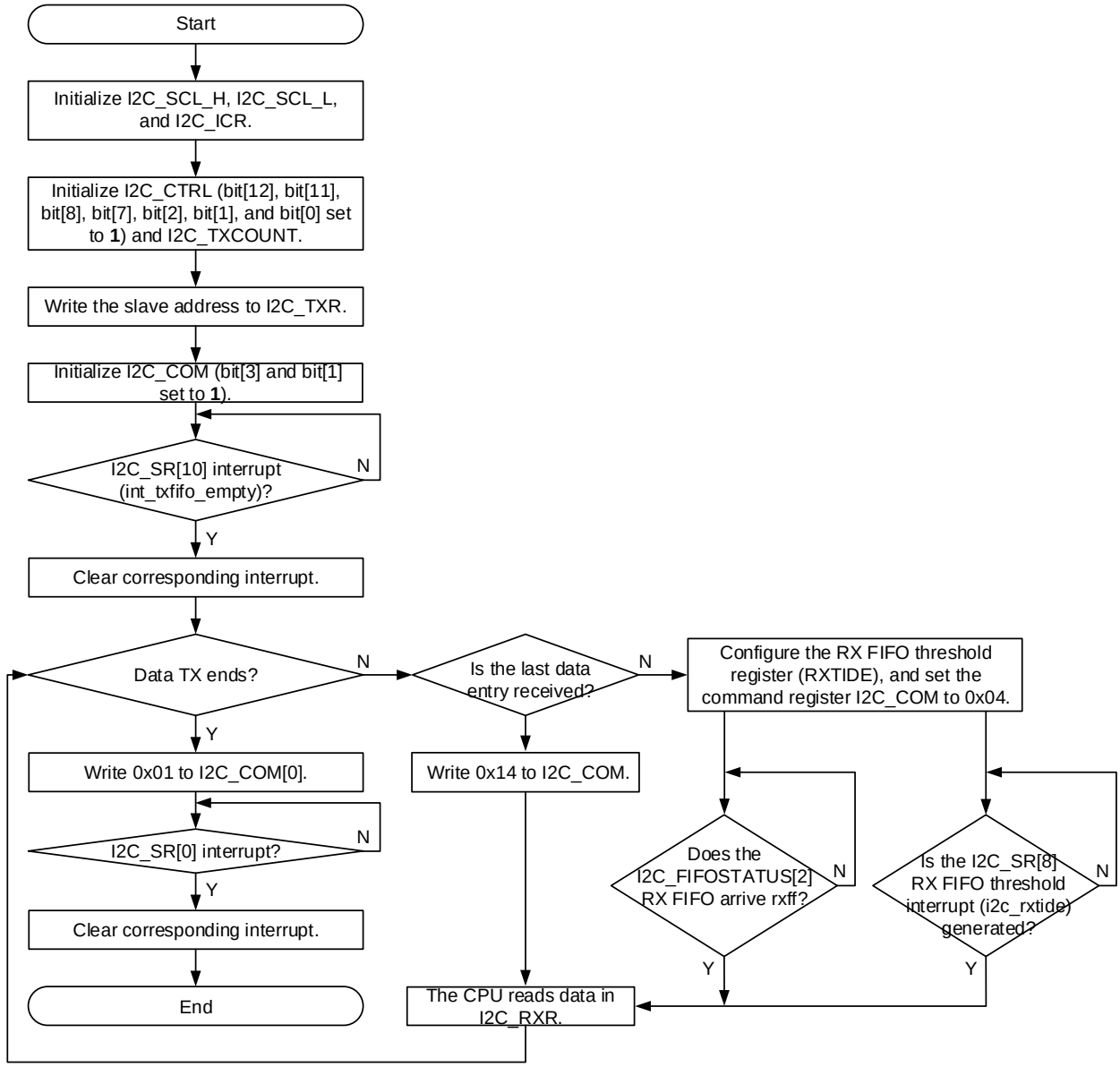
Figure 6-5 Data transmission (FIFO used)



Data Reception Process of the I²C Master

Figure 6-6 shows the process of receiving data by using the I²C master.

Figure 6-6 Data reception (FIFO used)



6.4.4 Register Summary

Table 6-12 describes I²C registers.

Table 6-12 Summary of I2C registers (base address of I2C0: 0x4001_8000; base address of I2C1: 0x4001_9000)

Offset Address	Register	Description	Page
0x00	I2C_CTRL	I²C control register	6-69
0x04	I2C_COM	I²C command register	6-70
0x08	I2C_ICR	I²C interrupt clear register	6-71

Offset Address	Register	Description	Page
0x0C	I2C_SR	I²C module status register	6-72
0x10	I2C_SCL_H	I²C SCL high-level cycle count register	6-74
0x14	I2C_SCL_L	I²C SCL low-level cycle count register	6-74
0x18	I2C_TXR	I²C TX data register	6-75
0x1C	I2C_RXR	I²C RX data register	6-75
0x20	I2C_FIFOSTATUS	FIFO status register	6-76
0x24	I2C_TXCOUNT	TX FIFO data count register	6-76
0x28	I2C_RXCOUNT	RX FIFO data count register	6-76
0x2C	I2C_RXTIDE	RX FIFO overflow threshold register	6-77
0x30	I2C_TXTIDE	TX FIFO overflow threshold register	6-77

6.4.5 Register Description

I2C_CTRL

I2C_CTRL is the I²C control register. It is used to control the I²C and interrupt mask.

Offset Address: 0x00 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:13]	-	reserved	Reserved	0x00000
[12]	RW	int_txfifo_over_mask	TX FIFO data completion interrupt mask 0: enabled 1: disabled	0x0
[11]	RW	mode_ctrl	I²C operating mode select 0: The FIFO transmission mode is not used. 1: The FIFO transmission mode is used.	0x0
[10]	RW	int_txtide_mask	TX FIFO overflow interrupt mask 0: enabled 1: disabled	0x0
[9]	RW	int_rxtide_mask	RX FIFO overflow interrupt mask 0: enabled 1: disabled	0x0
[8]	RW	i2c_en	I²C enable 0: disabled 1: enabled	0x0

Bit	Access	Name	Description	Reset
[7]	RW	int_mask	General I²C interrupt mask 0: enabled 1: disabled	0x0
[6]	RW	int_start_mask	TX completion interrupt mask for the start condition of the master 0: enabled 1: disabled	0x0
[5]	RW	int_stop_mask	TX completion interrupt mask for the stop condition of the master 0: enabled 1: disabled	0x0
[4]	RW	int_tx_mask	Master TX interrupt mask 0: enabled 1: disabled	0x0
[3]	RW	int_rx_mask	Master RX interrupt mask 0: enabled 1: disabled	0x0
[2]	RW	int_ack_err_mask	Slave ACK error interrupt mask 0: enabled 1: disabled	0x0
[1]	RW	int_arb_loss_mask	Bus arbitration failure interrupt mask 0: enabled 1: disabled	0x0
[0]	RW	int_done_mask	Bus transfer completion interrupt mask 0: enabled 1: disabled	0x0

I2C_COM

NOTICE

The corresponding interrupts must be cleared before or during system initialization. I2C_COM bit[3:0] are automatically cleared upon completion of the operation.

I2C_COM is the I²C command register. It is used to configure the commands for the working of the I²C module.

Offset Address: 0x04 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:5]	-	reserved	Reserved	0x0000000
[4]	RW	op_ack	Whether the master sends an ACK as a receiver 0: yes 1: no	0x0
[3]	RW	op_start	Generation of the start condition 0: The operation is complete. 1: The operation is valid.	0x0
[2]	RW	op_rd	Generation of the read operation 0: The operation is complete. 1: The operation is valid.	0x0
[1]	RW	op_we	Generation of the write operation 0: The operation is complete. 1: The operation is valid.	0x0
[0]	RW	op_stop	Generation of the stop condition 0: The operation is complete. 1: The operation is valid.	0x0

I2C_ICR

NOTICE

When a new interrupt is generated, the I²C module automatically clears the corresponding bit of I2C_ICR.

I2C_ICR is the I²C interrupt clear register.

Offset Address: 0x08 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:10]	-	reserved	Reserved	0x000000
[9]	WC	clr_int_txfifo_over	TX FIFO data completion interrupt clear 0: not cleared 1: cleared	0x0
[8]	WC	clr_int_txtide	TX FIFO overflow interrupt clear 0: not cleared 1: cleared	0x0

Bit	Access	Name	Description	Reset
[7]	WC	clr_int_rxtide	RX FIFO overflow interrupt clear 0: not cleared 1: cleared	0x0
[6]	WC	clr_int_start	TX completion interrupt clear for the start condition of the master 0: not cleared 1: cleared	0x0
[5]	WC	clr_int_stop	TX completion interrupt clear for the stop condition of the master 0: not cleared 1: cleared	0x0
[4]	WC	clr_int_tx	Master TX interrupt clear 0: not cleared 1: cleared	0x0
[3]	WC	clr_int_rx	Master RX interrupt clear 0: not cleared 1: cleared	0x0
[2]	WC	clr_int_ack_err	Slave ACK error interrupt clear 0: not cleared 1: cleared	0x0
[1]	WC	clr_int_arb_loss	Bus arbitration failure interrupt clear 0: not cleared 1: cleared	0x0
[0]	WC	clr_int_done	Bus transfer completion interrupt clear 0: not cleared 1: cleared	0x0

I2C_SR

NOTICE

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, and clear I2C_COM or write a new operation command to I2C_COM.

I2C_SR is the I²C module status register. It is used to read the operating status of the I²C module.

Offset Address: 0x0C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:11]	-	reserved	Reserved	0x000000
[10]	RO	int_txfifo_over	TX FIFO data completion interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[9]	RO	int_txtide	TX FIFO overflow interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[8]	RO	int_rxtide	RX FIFO overflow interrupt 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[7]	RO	bus_busy	Bus busy flag 0: idle 1: busy	0x0
[6]	RO	int_start	TX completion interrupt for the start condition of the master 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[5]	RO	int_stop	TX completion interrupt for the stop condition of the master 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[4]	RO	int_tx	Master TX interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[3]	RO	int_rx	Master RX interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[2]	RO	int_ack_err	Slave ACK error interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[1]	RO	int_arb_loss	Failure interrupt flag for the bus arbitration 0: No interrupt is generated. 1: An interrupt is generated.	0x0
[0]	RO	int_done	Bus transfer completion interrupt flag 0: No interrupt is generated. 1: An interrupt is generated.	0x0

I2C_SCL_H

NOTICE

During or before system initialization, you must set I2C_CTRL bit[7] to **0**.

I2C_SCL_H is the I²C SCL high-level cycle count register. It is used to configure the number of SCL high-level cycles when the I²C module is working.

Offset Address: 0x10 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	scl_h	Number of SCL high-level cycles. The actual number of SCL high-level cycles is 2 times the configured value.	0x0000

I2C_SCL_L

NOTICE

During or before system initialization, you must set I2C_CTRL bit[7] to **0**.

I2C_SCL_L is the I²C SCL low-level cycle count register. It is used to configure the number of SCL low level cycles when the I²C module is working.

Offset Address: 0x14 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	scl_l	Number of SCL low level cycles. The actual number of SCL low level cycles is 2 times the configured value.	0x0000

I2C_TXR

NOTICE

In non-FIFO mode, the I²C module does not modify the content of I2C_TXR after transmission is complete; in FIFO mode, the written data is automatically loaded to the transmitted FIFO for storage until the data transmission is complete.

I2C_TXR is the I²C TX data register. It is used to configure the data to be transmitted when the I²C module is working.

Offset Address: 0x18 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RW	i2c_txr	Data TX by the master	0x00

I2C_RXR

NOTICE

In non-FIFO mode, the I2C_RXR data is valid when I2C_SR bit[3] is **1**. The data is retained until the next read operation starts. In FIFO mode, reading I2C_RXR directly fetches data from the RX FIFO.

I2C_RXR is the I²C RX data register. It is used for the master to receive data from the slave.

Offset Address: 0x1C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:8]	-	reserved	Reserved	0x000000
[7:0]	RO	i2c_rxr	Data received by the master	0x00

I2C_FIFOSTATUS

I2C_FIFOSTATUS is the FIFO status register.

Offset Address: 0x20 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3]	RO	rxfe	RX FIFO empty status 0: not empty 1: empty	0x0
[2]	RO	rxff	RX FIFO full status 0: not full 1: full	0x0
[1]	RO	txfe	TX FIFO empty status 0: not empty 1: empty	0x0
[0]	RO	txff	TX FIFO full status 0: not full 1: full	0x0

I2C_TXCOUNT

I2C_TXCOUNT is the TX FIFO data count register.

Offset Address: 0x24 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x0000000
[5:0]	WC	txcount	Reading this register returns the number of characters in the TX FIFO. Writing any value to this register clears the TX FIFO.	0x00

I2C_RXCOUNT

I2C_RXCOUNT is the RX FIFO data count register.

Offset Address: 0x28 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x0000000
[5:0]	WC	rxcount	Reading this register returns the number of	0x00

Bit	Access	Name	Description	Reset
			characters in the RX FIFO. Writing any value to this register clears the RX FIFO.	

I2C_RXTIDE

I2C_RXTIDE is the RX FIFO overflow threshold register.

Offset Address: 0x2C Total Reset Value: 0x0000_0001

Bit	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x0000000
[5:0]	RW	rxtide	Value for triggering the int_rxtide interrupt When the number of characters in the RX FIFO is greater than or equal to I2C_RXTIDE[rxtide], the RX FIFO overflow interrupt is triggered.	0x01

I2C_TXTIDE

I2C_TXTIDE is the TX FIFO overflow threshold register.

NOTE
The characters in the TX FIFO are removed only after data is successfully transmitted.

Offset Address: 0x30 Total Reset Value: 0x0000_0001

Bit	Access	Name	Description	Reset
[31:6]	-	reserved	Reserved	0x0000000
[5:0]	RW	txtide	Value for triggering the int_txtide interrupt When the number of characters in the TX FIFO is less than or equal to I2C_TXTIDE[txtide], the TX FIFO overflow interrupt is triggered.	0x01

6.5 SPI

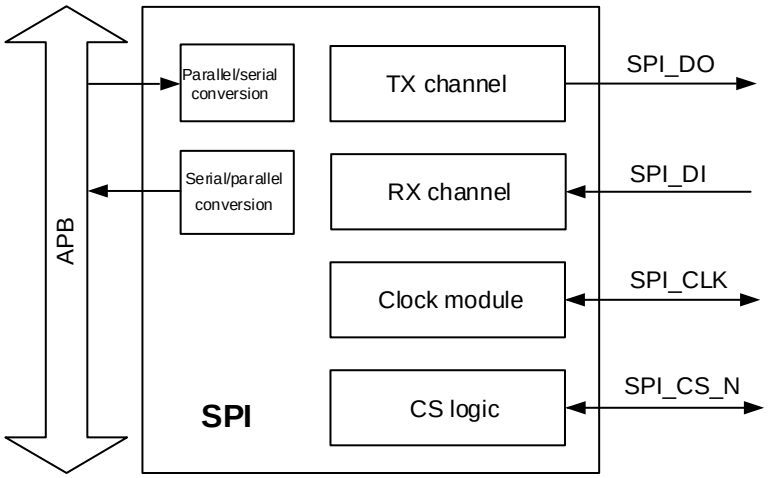
6.5.1 Overview

The serial peripheral interface (SPI) module implements serial-to-parallel conversion and parallel-to-serial conversion of data. The SPI module can act as a master or slave to

communicate with peripheral devices in synchronous serial mode. The peripheral devices must support the SPI frame format.

The SPI working reference clock of chips is 160 MHz. Figure 6-7 shows the functional block diagram of the SPI.

Figure 6-7 SPI functional block diagram



6.5.2 Features

The SPI has the following features:

- Supports programmable interface clock frequency.
  - As a master, supports maximum divide-by-4 clock frequency of the interface.
  - As a slave, supports maximum divide-by-8 clock frequency of the interface.
- Provides different FIFO specifications of SPI0 and SPI1.
  - SPI0: Supports FIFOs with the 256 x 16-bit word width (one TX FIFO and one RX FIFO).
  - SPI1: Supports FIFOs with the 64 x 16-bit word width (one TX FIFO and one RX FIFO).
- Supports the following SPI frame formats:
  - Motorola
  - Texas Instruments (TI)
  - National Microwire
- Supports programmable serial data frame length of 4-bit to 16-bit.
- Independently masks TX FIFO interrupts, RX FIFO interrupts, RX timeout interrupts, and RX FIFO overrun interrupts.
- Supports the internal loopback test mode.
- Supports the direct memory access (DMA) operation, but cannot serve as the flow control device of DMA.

6.5.3 Operating Modes

6.5.3.1 Interface Signals

Master mode

Table 6-13 lists the SPI interface signals.

Table 6-13 Description of SPI master interface signals

Signal	Width (Bit)	Direction	Description
SPI_CLK	1	O	SPI serial clock signal, generated and controlled by the master
SPI_CS_N	1	O	SPI chip select signal, active low
SPI_DI	1	I	Data input
SPI_DO	1	O	Data output

Slave mode

Table 6-14 lists the SPI interface signals.

Table 6-14 Description of SPI slave interface signals

Signal	Width (Bit)	Direction	Description
SPI_CLK	1	I	SPI serial clock signal, generated and controlled by the master
SPI_CS_N	1	I	SPI chip select signal, active low
SPI_DI	1	I	Data input
SPI_DO	1	O	Data output

6.5.3.2 Transfer Frame Format

The SPI can communicate with any serial master or slave device that supports the SPI frame formats.

NOTE

Acronyms and abbreviations in Figure 6-8 to Figure 6-15 are described as follows:
MSB: most significant bit
LSB: least significant bit
Q: undefined signal

SPO indicates the SPICLKOUT polarity, and SPH indicates the SPICLKOUT phase. The corresponding register bits are SPICR0 bit[7:6].

Motorola Frame Format

[SPO = 0, SPH = 0]

Figure 6-8 shows the format of an SPI single format.

Figure 6-8 Format of a single SPI frame (SPO = 0, SPH = 0)

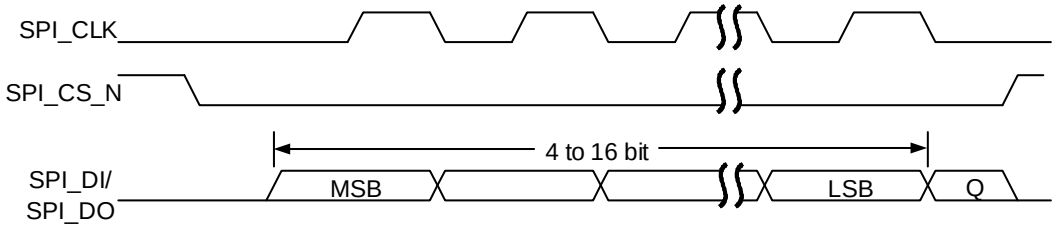
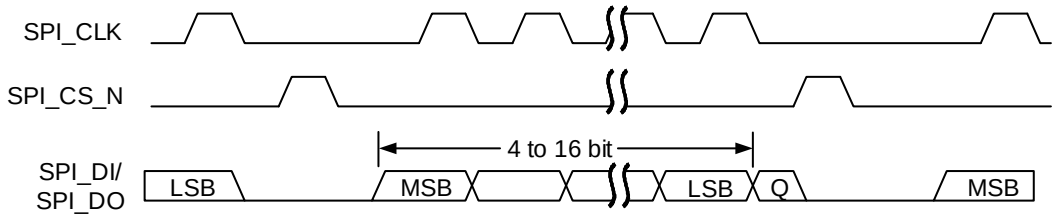


Figure 6-9 shows the format of consecutive SPI frames.

Figure 6-9 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 signal 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 signal to low starts data transfer. At this time, slave data is placed in the master RX data line SPI_DI. 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. Data is captured at the rising edge of the SPI_CLK clock and transmitted at the falling edge.

If a single word (4 – 16 bits) is transferred, SPI_CS_N is restored to high level one SPI_CLK cycle later after the last bit is captured.

If words are transferred consecutively, SPI_CS_N is pulled up by one clock cycle at the 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 (4 – 16 bits) transfer interval in consecutive transfer. When the consecutive transfer ends, SPI_CS_N is restored to high level one SPI_CLK cycle later after the last bit is captured.

[SPO = 0, SPH = 1]

Figure 6-10 shows the format of an SPI single format.

Figure 6-10 Format of a single SPI frame (SPO = 0, SPH = 1)

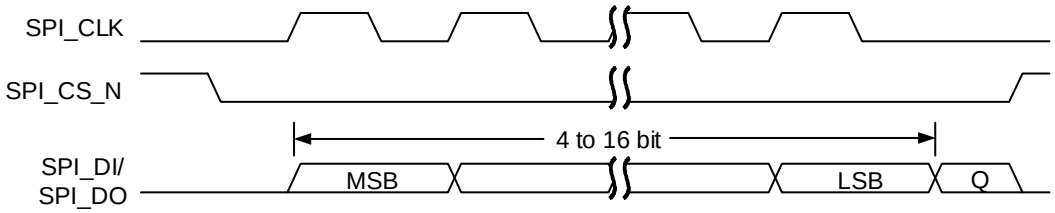
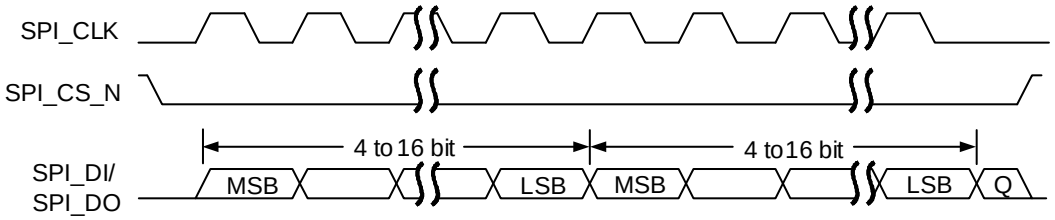


Figure 6-11 shows the format of consecutive SPI frames.

Figure 6-11 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 signal 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 data transfer. The slave data is immediately transmitted to the master RX data line SPI_DI. Half an SPI_CLK clock cycle later, the valid data of the master is valid on its own transmission line, and SPI_CLK generates the first rising edge. Data is captured at the falling edge of SPI_CLK and transmitted at the rising edge.

If a single word (4 – 16 bits) is transferred, SPI_CS_N is restored to high level one SPI_CLK cycle later after the last bit is captured.

If words are transferred consecutively, 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 bit is captured.

[SPO = 1, SPH = 0]

Figure 6-12 shows the format of an SPI single format.

Figure 6-12 Format of a single SPI frame (SPO = 1, SPH = 0)

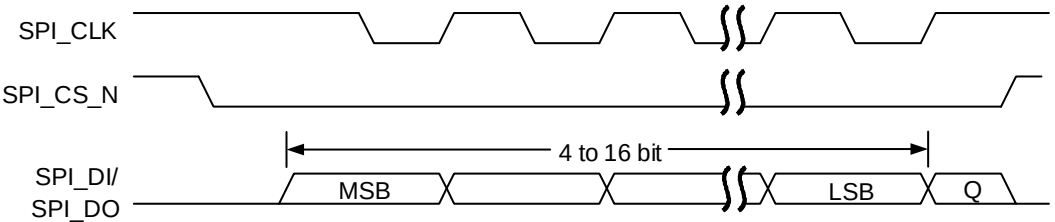
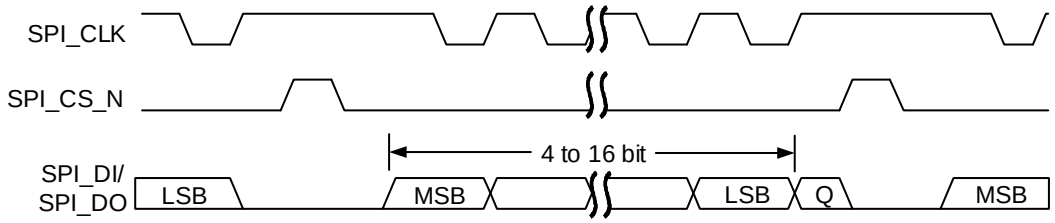


Figure 6-13 shows the format of consecutive SPI frames.

Figure 6-13 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 signal 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 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. Half an SPI_CLK clock cycle later, the SPI_CLK master pin is set to low, indicating that data is captured on the falling edge of the SPI_CLK clock and transmitted on the rising edge of the SPI_CLK clock.

If a single word (4 – 16 bits) is transferred, SPI_CS_N is restored to high level one SPI_CLK cycle later after the last bit is captured.

If words are transferred consecutively, SPI_CS_N must be pulled up at each word transfer interval. The reason is that when SPH is **0**, the slave select pin retains the data in the internal serial device register. SPI_CS_N is restored to high level one SPI_CLK clock cycle later after the last bit is captured.

[SPO = 1, SPH = 1]

Figure 6-14 shows the format of an SPI single format.

Figure 6-14 SPI single frame format (SPO = 1, SPH = 1)

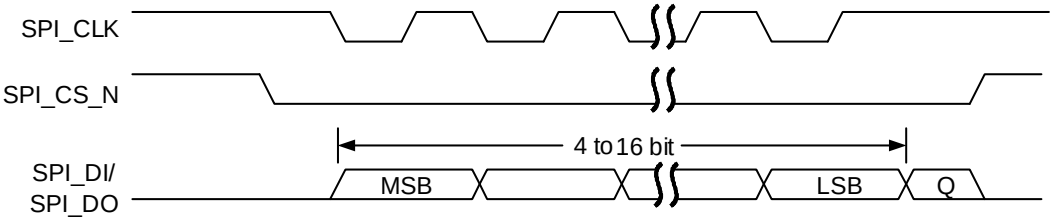
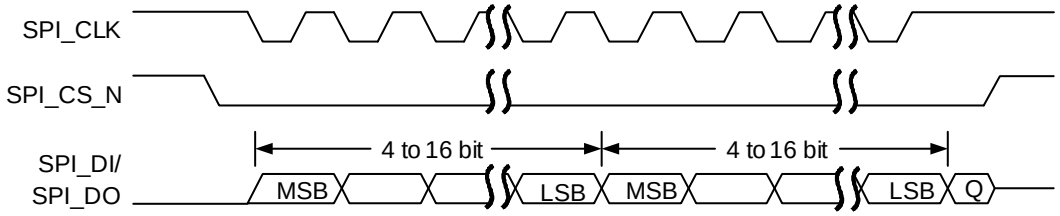


Figure 6-15 shows the format of consecutive SPI frames.

Figure 6-15 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 signal 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 data transfer. The slave data is immediately transmitted to the master RX data line SPI_DI. Half an SPI_CLK clock cycle later, the master data is valid on its own transmission line, and SPI_CLK generates the first falling edge. Data is captured at the rising edge of the SPI_CLK clock and transmitted at the falling edge.

If a single word (4 – 16 bits) is transferred, SPI_CS_N is restored to high level one SPI_CLK cycle later after the last bit is captured.

If words are transferred consecutively, the SPI_CS_N signal retains low during data transfer. SPI_CS_N is restored to high level one SPI_CLK cycle later after the last bit is captured. The end mode is the same as that during single word transfer.

6.5.3.3 Working Modes

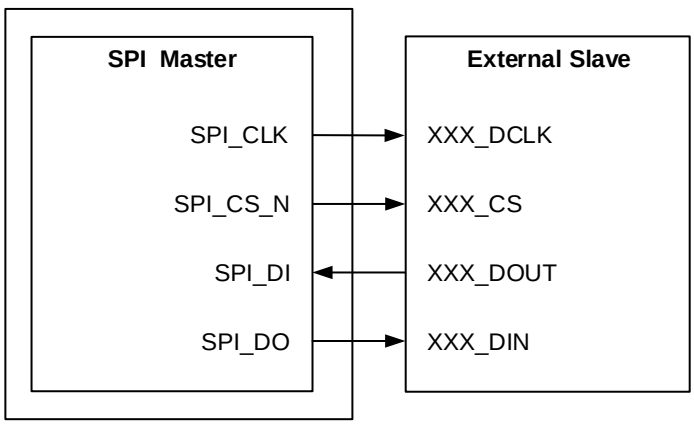
The SPI module supports data transfer in interrupt mode or query mode.

The SPI module supports the master and slave working modes.

Master Mode

In master mode, the SPI module is initialized and directly controls all peripherals. Figure 6-16 shows the communications between the SPI module configured as the master and the peripheral configured as the slave.

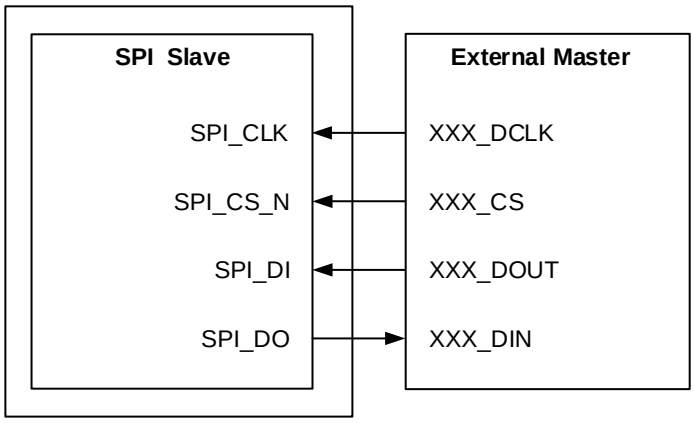
Figure 6-16 SPI connected to a single slave



Slave Mode

In slave mode, the SPI module is initialized and controlled by the master on the serial bus. Figure 6-17 shows the communications between the SPI module configured as the slave and the peripheral configured as the master.

Figure 6-17 SPI connected to a single master



6.5.3.4 Interrupt Processing

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. This interrupt is enabled when data in the RX FIFO reaches the threshold or there is more valid data.

- SPITXINTR

  TX FIFO interrupt. This interrupt is enabled when data in the TX FIFO reaches the threshold or there is less valid data.

- SPIRORINTR

  RX overflow interrupt. When the FIFO is full and new data needs to be written to the FIFO, an FIFO overrun 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.

- 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.

6.5.3.5 Description

Transmitting data in interrupt or query mode is used as an example.

Initialization

Perform the following initialization steps:

Step 1 Write **0** to SPICR1[SSE] to disable the SPI.

Step 2 Configure SPICR0 to set the frame format, bit width of the transferred data, SPICLKOUT phase, and polarity.

Step 3 Configure SPICPSR to specify the required clock divider.

Step 4 In interrupt mode, set SPIIMSC to mask the corresponding interrupt signal. In query mode, masked all generated interrupt signals.

----End

Transmitting Data

To transmit fixed-length data (for example, four data entries), perform the following steps:

Step 1 Write **1** to SPICR1[sse] to enable the SPI.

Step 2 Read SPISR to check whether the FIFO is empty or full. If SPISR[tnf] is 1, the TX FIFO is not full. Write four data entries to SPIDR. If the FIFO is full, wait until the previous data is transmitted and do not input data until the FIFO is empty.

Step 3 Read SPISR to check whether the FIFO is empty or full until all data is transmitted.

Step 4 Write 0 to SPICR1[sse] to disable the SPI.

----End

Receiving Data

To receive fixed-length data (for example, four data entries), perform the following steps:

Step 1 Write **1** to SPICR1[sse] to enable the SPI.

Step 2 Read SPISR to check whether the RX FIFO is empty or full. If the RX FIFO is not empty (SPISR[rne]=1), read SPIDR to receive four data entries.

Step 3 Query the status register SPISR[bsy] until the FIFO is not busy.

Step 4 Write 0 to SPICR1[sse] to disable the SPI.

----End

6.5.4 Register Summary

Table 6-15 lists the SPI registers.

Table 6-15 Summary of SPI registers (base address of SPI0: 0x4005_8000; base address of SPI1: 0x4005_9000)

Offset Address	Register	Description	Page
0x000	SPICR0	SPI control register 0	6-86

Offset Address	Register	Description	Page
0x004	SPICR1	SPI control register 1	6-87
0x008	SPIDR	SPI TX/RX data register	6-88
0x00C	SPISR	SPI status register	6-88
0x010	SPICPSR	SPI clock frequency division register	6-89
0x014	SPIIMSC	SPI interrupt mask register	6-89
0x018	SPIRIS	SPI raw interrupt status register	6-90
0x01C	SPIMIS	SPI masked interrupt status register	6-90
0x020	SPIICR	SPI interrupt clear register	6-91
0x024	SPIDMACR	SPI DMA enable register	6-91
0x028	SPITXFIFOCR	SPI TX FIFO control register	6-92
0x02C	SPIRXFIFOCR	SPI RX FIFO control register	6-93

6.5.5 Register Description

SPICR0

SPICR0 is SPI control register 0.

Offset Address: 0x000 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[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: SSPCLKOUT = FSSPCLK/[CPSDVSR x (1 + SCR)] Where, **CPSDVSR** is an even number ranging from 2 to 254, and is specified by configuring SPICPSR. **Note: The following conditions must be met:** **Frequency divider [CPSDVSR x (1 + SCR)] of the slave device ≥ 8** **Data transmission rate of the master device (SSPCLKOUT) ≤ Data transmission rate of the slave device**	0x00
[7]	RW	sph	SPICLKOUT phase For details, see the SPI frame format in the *Hi3861/Hi3861L/Hi3881 V100 Wi-Fi Chip Data Sheet (Hardware)*.	0x0

Bit	Access	Name	Description	Reset
[6]	RW	spo	SPICLKOUT polarity For details, see the SPI frame format in the *Hi3861/Hi3861L/Hi3881 V100 Wi-Fi Chip Data Sheet (Hardware)*.	0x0
[5:4]	RW	frf	Frame format select 00: Motorola SPI frame format 01: TI synchronous serial frame 10: National Microwire frame format 11: Reserved	0x0
[3:0]	RW	dss	Data bit width 0x3: 4 bits 0x4: 5 bits 0x5: 6 bits 0x6: 7 bits 0x7: 8 bits 0x8: 9 bits 0x9:10 bits 0xA: 11 bits 0xB: 12 bits 0xC: 13 bits 0xD: 14 bits 0xE: 15 bits 0xF: 16 bits Other values: reserved	0x0

SPICR1

SPICR1 is SPI control register 1.

Offset Address: 0x004 Total Reset Value: 0x7F00

Bit	Access	Name	Description	Reset
[15:5]	-	reserved	Reserved	0x3F8
[4]	RW	bigend	Data endian mode 0: little endian 1: big endian	0x0
[3]	-	reserved	Reserved	0x0
[2]	RW	ms	Master or slave mode 0: master mode (default value)	0x0

			1: slave mode **Note: This bit can be changed only when the SPI is disabled.**	
[1]	RW	sse	SPI enable 0: disabled 1: enabled	0x0
[0]	RW	lbm	1: outloop 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.	0x0

SPIDR

SPIDR is the SPI TX/RX data register.

Offset Address: 0x008 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:0]	RW	data	TX or RX FIFO Read: RX FIFO Write: TX FIFO If the data is less than 16 bits, data must be right-aligned. The TX logic ignores the unused upper bits, and the RX logic automatically aligns data to the right.	0x0000

SPISR

SPISR is the SPI status register.

Offset Address: 0x00C Total Reset Value: 0x0003

Bit	Access	Name	Description	Reset
[15:5]	-	reserved	Reserved	0x000
[4]	RO	bsy	SPI busy indicator 0: idle 1: busy	0x0
[3]	RO	rff	Whether the RX FIFO is full 0: no 1: full	0x0
[2]	RO	rne	Whether the RX FIFO is empty 0: yes	0x0

			1: no	
[1]	RO	tnf	Whether the TX FIFO is full 0: full 1: not full	0x1
[0]	RO	tfe	Whether the TX FIFO is empty 0: no 1: yes	0x1

SPICPSR

SPICPSR is the SPI clock frequency division register.

Offset Address: 0x010 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x00
[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 less significant bit (LSB) is read as **0**.	0x00

SPIIMSC

SPIIMSC is the SPI interrupt mask register.

Offset Address: 0x014 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x000
[3]	RW	txim	Interrupt mask when the TX FIFO is half empty or less 0: enabled 1: disabled	0x0
[2]	RW	rxim	Interrupt mask when the RX FIFO is half empty or less 0: enabled 1: disabled	0x0
[1]	RW	rtim	RX timeout interrupt mask 0: enabled 1: disabled	0x0

[0]	RW	rorim	RX FIFO overflow interrupt mask 0: enabled 1: disabled When the value is **1**, the hardware flow control function is enabled. That is, when the RX FIFO is full, the SPI stops transmitting data.	0x0

SPIRIS

SPIRIS is the raw interrupt status register.

Offset Address: 0x018 Total Reset Value: 0x0008

Bit	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x000
[3]	RO	txris	Raw TX FIFO interrupt status 0: No interrupt is generated. 1: The interrupt is generated.	0x1
[2]	RO	rxris	Raw RX FIFO interrupt status 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[1]	RO	rtris	Raw RX timeout interrupt status 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[0]	RO	rorris	Raw RX overflow interrupt status 0: No interrupt is generated. 1: The interrupt is generated.	0x0

SPIMIS

SPIMIS is the SPI masked interrupt status register.

Offset Address: 0x01C Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x000
[3]	RO	txmis	Masked TX FIFO interrupt status 0: No interrupt is generated. 1: The interrupt is generated.	0x0

Bit	Access	Name	Description	Reset
[2]	RO	rxmis	Masked RX FIFO interrupt status 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[1]	RO	rtmis	Masked RX timeout interrupt state 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[0]	RO	rormis	Masked RX overflow interrupt state 0: No interrupt is generated. 1: The interrupt is generated.	0x0

SPIICR

SPIICR is the SPI interrupt clear register.

Offset Address: 0x020 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:2]	-	reserved	Reserved	0x0000
[1]	WC	rtic	RX timeout interrupt clear Writing 0 has no effect. Writing 1 clears the interrupt.	0x0
[0]	WC	roric	RX overflow interrupt clear 0: invalid 1: interrupt cleared	0x0

SPIDMACR

SPIDMACR is an SPI DMA enable register.

Offset Address: 0x024 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:2]	-	reserved	Reserved	0x0000
[1]	RW	dmatxen	SPI DMA TX enable 0: disabled 1: enabled	0x0
[0]	RW	dmarxen	SPI DMA RX enable 0: disabled	0x0

			1: enabled	

SPITXFIFOCR

SPITXFIFOCR is the SPI TX FIFO control register.

Offset Address: 0x028 Total Reset Value: 0x0009

Bit	Access	Name	Description	Reset
[15:6]	-	reserved	Reserved	0x000
[5:3]	RW	txintsize	Threshold for triggering a TX FIFO interrupt. That is, when the number of data entries in the TX FIFO is less than or equal to the configured value of [txintsize], [txris] is valid. SPI0: 000: 1 001: 4 010: 8 011: 16 100: 32 101: 64 110: 128 111: 192 SPI1: 000: 1 001: 4 010: 8 011: 16 100: 32 101: 48 110: 56 111: 64	0x1
[2:0]	RW	dmatxbrsizepclk	Configures the burst request of the TX FIFO. Based on the configured threshold level, when the data in the TX FIFO reaches the corresponding level, a DMA burst request is generated. SPI0: 000: The threshold level is 255. 001: The threshold level is 252. 010: The threshold level is 248. 011: The threshold level is 240. 100: The threshold level is 224.	0x1

Bit	Access	Name	Description	Reset
			101: The threshold level is 192. 110: The threshold level is 128. 111: The threshold level is 64. SPI1: 000: The threshold level is 63. 001: The threshold level is 60. 010: The threshold level is 48. 011: The threshold level is 32. 100: The threshold level is 16. 101: The threshold level is 8. 110: The threshold level is 4. 111: The threshold level is 1.	

SPIRXFIFOCR

SPIRXFIFOCR is the SPI RX FIFO control register.

Offset Address: 0x02C Total Reset Value: 0x0009

Bit	Access	Name	Description	Reset
[15:6]	-	reserved	Reserved	0x000
[5:3]	RW	rxintsize	Threshold for triggering an RX FIFO interrupt. When the number of data entries in the RX FIFO is greater than or equal to **RXINTSize**, **RXRIS** is valid and the word length is 16 bits. SPI0: 000: 255 001: 252 010: 248 011: 240 100: 224 101: 192 110: 128 111: 32 SPI1: 000: 65 001: 62 010: 48 011: 32 100: 16	0x1

Bit	Access	Name	Description	Reset
			101: 8 110: 4 111: 1	
[2:0]	RW	dmarxbrsizepclk	Configures the burst request of the RX FIFO. Based on the configured threshold level, when the number of data entries in the RX FIFO reaches the corresponding level, a DMA burst request is generated. SPI0: 000: The threshold level is 1. 001: The threshold level is 4. 010: The threshold level is 8. 011: The threshold level is 16. 100: The threshold level is 32. 101: The threshold level is 64. 110: The threshold level is 128. 111: The threshold level is 192. SPI1: 000: The threshold level is 1. 001: The threshold level is 4. 010: The threshold level is 8. 011: The threshold level is 16. 100: The threshold level is 32. 101–111: The threshold level is 64.	0x1

6.6 PWM

6.6.1 Overview

The PWM module is used to generate PWM signals and adjust the illuminator brightness.

6.6.2 Features

The PWM mode has the following features:

- Integrates six PWM signals.
- Separately gates each PWM clock.
- Be able to divide frequency of the crystal clock (40/24 MHz) or 160 MHz by 1–65535.
- Adjusts the PWM signal duty cycle duty ratio from 1 to 1/65535.

6.6.3 Operating Modes

Two clocks are available for the PWM module: crystal clock (40/24 MHz) and 160 MHz.

Dividing the frequency of the 160 MHz clock by 25 times with the duty cycle of 1/5 is used as an example. Perform the following steps:

Step 1 Set CLK_SEL[pwm_clk_sel] to **0** to select the 160 MHz clock as the PWM clock.

Step 2 Set PWM_EN[pwm_en] to **1** to enable the PWM signal output.

Step 3 Set PWM_FREQ[pwm_freq] to **0x19** to set the frequency divider to 25.

Step 4 Set PWM_DUTY[pwm_duty] to **0x5** to set the duty cycle of the PWM signal to PWM_DUTY**[pwm_duty]**/PWM_FREQ**[pwm_freq] = 1/5**.

Step 5 set PWM_START[pwm_start] to **1** to make the configurations of Step 2, Step 3, and Step 4 to take effect.

----End

6.6.4 Register Summary

Table 6-16 describes PWM registers.

Table 6-16 Summary of PWM registers (PWM0 base address: 0x4004_0000; PWM1 base address: 0x4004_0100; PWM2 base address: 0x4004_0200; PWM3 base address: 0x4004_0300; PWM4 base address: 0x4004_0400; PWM5 base address: 0x4004_0500)

Offset Address	Register	Description	Page
0x00	PWM_EN	PWM enable register	
0x04	PWM_START	PWM configuration enforce register	
0x08	PWM_FREQ	PWM frequency control count value register	
0x0C	PWM_DUTY	PWM duty cycle count value register	

6.6.5 Register Description

PWM_EN

PWM_EN is the PWM enable register.

Offset Address: 0x00 Total Reset Value: 0x0000_0001

Bit	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	pwm_en	PWM function enable 0: disabled. The pwm_out is always **0**.	0x1

Bit	Access	Name	Description	Reset
			1: enabled	

PWM_START

PWM_START is the PWM configuration enforce register.

Offset Address: 0x04 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	pwm_start	PWM configuration enforce register. 0: disabled 1: enabled. The logic automatically clears this bit after the previously assigned PWM registers take effect.	0x0

PWM_FREQ

PWM_FREQ is the PWM frequency control count value register.

Offset Address: 0x08 Total Reset Value: 0x0000_05DC

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	pwm_freq	PWM frequency control count value, that is, the frequency divider for the PWM clock. The value ranges from 1 to 165535.	0x05DC

PWM_DUTY

PWM_DUTY is the PWM duty cycle count value register.

Offset Address: 0x0C Total Reset Value: 0x0000_02EE

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	pwm_duty	PWM duty cycle control count value. The value ranges from 1 to 65535. The ratio of pwm_duty to pwm_freq is the duty cycle.	0x02EE

6.7 HPM

6.7.1 Overview

HPM is a hardware performance monitoring module. It dynamically adjusts the power consumption of different corner chips by using the selective voltage binning (SVB) technology.

6.7.2 Features

The HPM module has the following features:

- Integrates three processes: standard voltage threshold (SVT), high voltage threshold (HVT), and low voltage threshold (LVT).
- Supports clock divider configuration.
- Supports cyclic monitoring and one-time monitoring.

6.7.3 Operating Modes

The chip HPM module integrates three processes (SVT, HVT, and LVT). To use the SVT process of the HPM module, perform the following steps:

Step 1 Set PERI_PMC22[hpm0_div] to **1** to select the 48 MHz clock divided by 2 as the HPM working clock.

Step 2 Set PERI_PMC22[hpm0_monitor_en] to **1** to enable the HPM module.

Step 3 Read PERI_PMC23[hpm0_pc_record0], [hpm0_pc_record1], and PERI_PMC24[hpm0_pc_record2], and [hpm0_pc_record3].

Step 4 Average the four read values in step 3, and the average value is the final original code word. A larger value indicates better hardware performance (oscillator frequency). This value is used to monitor the chip performance.

----End

6.7.4 Register Summary

Table 6-17 describes the HMP registers.

Table 6-17 Summary of HMP registers (base address: 0x4006_8000)

Offset Address	Register	Description	Page
0x0058	PERI_PMC22	HPM0 clock and soft reset control register	6-98
0x005C	PERI_PMC23	HPM0 original code word and alarm information register	6-99
0x0060	PERI_PMC24	HPM0 original code word and RCC code report register	6-100
0x0064	PERI_PMC25	HPM0 original code pattern threshold configuration register	6-100

Offset Address	Register	Description	Page
0x0068	PERI_PMC26	HPM1 clock and soft reset control register	6-101
0x006C	PERI_PMC27	HPM1 original code word and alarm information register	6-102
0x0070	PERI_PMC28	HPM1 original code word and RCC code report register	6-102
0x0074	PERI_PMC29	HPM1 original code pattern threshold configuration register	6-103
0x0078	PERI_PMC30	HPM2 clock and soft reset control register	6-103
0x007C	PERI_PMC31	HPM2 original code word and alarm information register	6-104
0x0080	PERI_PMC32	HPM2 original code word and RCC code report register	6-105
0x0084	PERI_PMC33	HPM2 original code pattern threshold configuration register	6-105

6.7.5 Register Description

PERI_PMC22

PERI_PMC22 is the HPM0 clock and soft reset control register.

Offset Address: 0x0058 Total Reset Value: 0x0000_000A

Bit	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27]	RW	hpm0_rst_req	HPM0 reset request enable 0: disabled 1: enabled	0x0
[26]	RW	hpm0_monitor_en	HPM0 cyclic monitoring enable 0: disabled 1: enabled	0x0
[25]	RW	hpm0_bypass	Single-time HPM0 enable signal 0: The start signal of HPM0 is determined by [hpm0_monitor_en]. 1: The start signal of the HPM0 is determined by [hpm0_en].	0x0
[24]	RW	hpm0_en	Single HPM0 measurement enable 0: The value is retained **0** before a process is started.	0x0

Bit	Access	Name	Description	Reset
			1: A frequency modulation process is started.	
[23:22]	RW	reserved	Reserved	0x0
[21:12]	RW	hpm0_offset	Offset after the HPM0 original code word is shifted rightwards	0x000
[11:10]	RW	reserved	Reserved	0x0
[9:8]	RW	hpm0_shift	Number of right shift bits of the HPM0 original code word	0x0
[7:6]	RW	reserved	Reserved	0x0
[5:0]	RW	hpm0_div	HPM0 clock divider The clock divider value is **[hpm0_div] + 1**. Supports frequency division by 2–64.	0x0A

PERI_PMC23

PERI_PMC23 is the HPM0 original code word and alarm information register.

Offset Address: 0x005C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25]	RO	hpm0_up_warning	Report flag when the HPM0 original code word exceeds the upper threshold 0: The value does not exceed the upper threshold. 1: The value exceeds the upper threshold.	0x0
[24]	RO	hpm0_low_warning	Report flag when the HPM0 original code word is lower than the lower threshold. 0: The value is not lower than the lower threshold. 1: The value is lower than the lower threshold.	0x0
[23:22]	-	reserved	Reserved	0x0
[21:12]	RO	hpm0_pc_record1	Reported value of the HPM0 original code word (the last historical value)	0x000
[11]	-	reserved	Reserved	0x0
[10]	RO	hpm0_pc_valid	HPM0 output validity indicator 0: the last read value of [hpm0_pc_record0] 1: the current read value of [hpm0_pc_record0]	0x0

Bit	Access	Name	Description	Reset
[9:0]	RO	hpm0_pc_record0	Reported value of the HPM0 original code word (current value)	0x000

PERI_PMC24

PERI_PMC24 is the HPM0 original code word and RCC code report register.

Offset Address: 0x0060 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x0
[28:24]	RO	hpm0_rcc	RCC code output by the HPM0	0x00
[23:22]	-	reserved	Reserved	0x0
[21:12]	RO	hpm0_pc_record3	Reported value of the HPM0 original code word (last three historical values)	0x000
[11:10]	-	reserved	Reserved	0x0
[9:0]	RO	hpm0_pc_record2	Reported value of the HPM0 original code word (last two historical values)	0x000

PERI_PMC25

PERI_PMC25 is the HPM0 original code pattern threshold configuration register.

Offset Address: 0x0064 Total Reset Value: 0x0100_0000

Bit	Access	Name	Description	Reset
[31:24]	RW	hpm0_monitor_period	HPM0 cyclic monitoring period If the configured value is **N**, the monitoring period (T) is as follows: T = N x 2048/1000 ms. Where, the maximum value of **N** is **0xFF**, the maximum monitoring interval is 522 ms, and the minimum monitoring interval is 2 ms.	0x01
[23]	RW	reserved	Reserved	0x0
[22:12]	RW	hpm0_lowlimit	Lower limit of the HPM0 original code pattern	0x000
[11:10]	RW	reserved	Reserved	0x0
[9:0]	RW	hpm0_uplimit	Upper limit of the HPM0 original code pattern	0x000

PERI_PMC26

PERI_PMC26 is the HPM1 clock and soft reset control register.

Offset Address: 0x0068 Total Reset Value: 0x0000_000A

Bit	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27]	RW	hpm1_rst_req	HPM1 reset request enable 0: disabled 1: enabled	0x0
[26]	RW	hpm1_monitor_en	HPM1 cyclic monitoring enable 0: disabled 1: enabled	0x0
[25]	RW	hpm1_bypass	Single-time HPM1 enable signal 0: The start signal of HPM1 is determined by [hpm1_monitor_en]. 1: The start signal of the HPM1 is determined by [hpm1_en].	0x0
[24]	RW	hpm1_en	Single HPM1 measurement enable 0: The value is retained **0** before a process is started. 1: A frequency modulation process is started.	0x0
[23:22]	RW	reserved	Reserved	0x0
[21:12]	RW	hpm1_offset	Offset after the HPM1 original code word is shifted rightwards	0x000
[11:10]	RW	reserved	Reserved	0x0
[9:8]	RW	hpm1_shift	Number of right shift bits of the HPM1 original code word	0x0
[7:6]	RW	reserved	Reserved	0x0
[5:0]	RW	hpm1_div	HPM1 clock divider The clock divider value is **[hpm1_div] + 1**. Supports frequency division by 2–64.	0x0A

PERI_PMC27

PERI_PMC27 is the HPM1 original code word and alarm information register.

Offset Address: 0x006C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25]	RO	hpm1_up_warning	Report flag when the HPM1 original code word exceeds the upper threshold. 0: The value does not exceed the upper threshold. 1: The value exceeds the upper threshold.	0x0
[24]	RO	hpm1_low_warning	Report flag when the HPM1 original code word is lower than the lower threshold 0: The value is not lower than the lower threshold. 1: The value is lower than the lower threshold.	0x0
[23:22]	-	reserved	Reserved	0x0
[21:12]	RO	hpm1_pc_record1	Reported value of the HPM1 original code word (last historical value)	0x000
[11]	-	reserved	Reserved	0x0
[10]	RO	hpm1_pc_valid	HPM1 output validity indicator 0: the last read value of [hpm1_pc_record0] 1: the current read value of [hpm1_pc_record0]	0x0
[9:0]	RO	hpm1_pc_record0	Reported value of the HPM1 original code word (current value).	0x000

PERI_PMC28

PERI_PMC28 is the HPM1 original code word and RCC code report register.

Offset Address: 0x0070 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x0
[28:24]	RO	hpm1_rcc	RCC code output by HPM1	0x00
[23:22]	-	reserved	Reserved	0x0
[21:12]	RO	hpm1_pc_record3	Reported value of the HPM1 original code word (last three historical values)	0x000
[11:10]	-	reserved	Reserved	0x0

Bit	Access	Name	Description	Reset
[9:0]	RO	hpm1_pc_record2	Reported value of the HPM1 original code word (last two historical values)	0x000

PERI_PMC29

PERI_PMC29 is the HPM1 original code pattern threshold configuration register.

Offset Address: 0x0074 Total Reset Value: 0x0100_0000

Bit	Access	Name	Description	Reset
[31:24]	RW	hpm1_monitor_period	HPM1 cyclic monitoring period If the configured value is **N**, the monitoring period (T) is as follows: T = N x 2048/1000 ms. Where, the maximum value of **N** is **0xFF**, the maximum monitoring interval is 522 ms, and the minimum monitoring interval is 2 ms.	0x01
[23]	RO	reserved	Reserved	0x0
[22:12]	RW	hpm1_lowlimit	Lower limit of the HPM1 original code pattern	0x000
[11:10]	RW	reserved	Reserved	0x0
[9:0]	RW	hpm1_uplimit	Upper limit of the HPM1 original code pattern	0x000

PERI_PMC30

PERI_PMC30 is the HPM2 clock and soft reset control register.

Offset Address: 0x0078 Total Reset Value: 0x0000_000A

Bit	Access	Name	Description	Reset
[31:28]	-	reserved	Reserved	0x0
[27]	RW	hpm2_rst_req	HPM2 reset request enable 0: disabled 1: enabled	0x0
[26]	RW	hpm2_monitor_en	HPM2 cyclic monitoring enable 0: disabled 1: enabled	0x0
[25]	RW	hpm2_bypass	Single-time HPM2 enable signal 0: The start signal of HPM2 is determined by [hpm2_monitor_en].	0x0

Bit	Access	Name	Description	Reset
			1: The start signal of the HPM2 is determined by [hpm2_en].	
[24]	RW	hpm2_en	Single HPM2 measurement enable 0: The value is retained **0** before a process is started. 1: A frequency modulation process is started.	0x0
[23:22]	RW	reserved	Reserved	0x0
[21:12]	RW	hpm2_offset	Offset after the HPM2 original code word is shifted rightwards	0x000
[11:10]	RW	reserved	Reserved	0x0
[9:8]	RW	hpm2_shift	Number of right shift bits of the original HPM2 code word	0x0
[7:6]	RW	reserved	Reserved	0x0
[5:0]	RW	hpm2_div	HPM2 clock divider The divider value is **[hpm2_div] + 1**. Supports frequency division by 2–64.	0x0A

PERI_PMC31

PERI_PMC31 is the HPM2 original code word and alarm information register.

Offset Address: 0x007C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25]	RO	hpm2_up_warning	Report flag when the HPM2 original code word exceeds the upper threshold 0: The value does not exceed the upper threshold. 1: The value exceeds the upper threshold.	0x0
[24]	RO	hpm2_low_warning	Report flag when the HPM2 original code word is lower than the lower threshold 0: The value is not lower than the lower threshold. 1: The value is lower than the lower threshold.	0x0
[23:22]	-	reserved	Reserved	0x0
[21:12]	RO	hpm2_pc_record1	Reported value of the HPM2 original code word (last historical value)	0x000
[11]	-	reserved	Reserved	0x0

Bit	Access	Name	Description	Reset
[10]	RO	hpm2_pc_valid	HPM2 output validity indicator 0: the last read value of [hpm2_pc_record0] 1: the current read value of [hpm2_pc_record0]	0x0
[9:0]	RO	hpm2_pc_record0	Reported value of the HPM2 original code word (current value)	0x000

PERI_PMC32

PERI_PMC32 is the HPM2 original code word and RCC code report register.

Offset Address: 0x0080 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:29]	-	reserved	Reserved	0x0
[28:24]	RO	hpm2_rcc	RCC code output by HPM2	0x00
[23:22]	-	reserved	Reserved	0x0
[21:12]	RO	hpm2_pc_record3	Reported value of the HPM2 original code word (last three historical values)	0x000
[11:10]	-	reserved	Reserved	0x0
[9:0]	RO	hpm2_pc_record2	Reported value of the HPM2 original code word (last two historical values)	0x000

PERI_PMC33

PERI_PMC33 is the HPM2 original code pattern threshold configuration register.

Offset Address: 0x0084 Total Reset Value: 0x0100_0000

Bit	Access	Name	Description	Reset
[31:24]	RW	hpm2_monitor_period	HPM2 cyclic monitoring period If the configured value is **N**, the monitoring period (T) is as follows: T = N x 2048/1000 ms. Where, the maximum value of **N** is **0xFF**, the maximum monitoring interval is 522 ms, and the minimum monitoring interval is 2 ms.	0x01
[23]	RO	reserved	Reserved	0x0
[22:12]	RW	hpm2_lowlimit	Lower limit of the HPM2 original code pattern	0x000

Bit	Access	Name	Description	Reset
[11:10]	RW	reserved	Reserved	0x0
[9:0]	RW	hpm2_uplimit	Upper limit of the HPM2 original code pattern	0x000

6.8 TSensor

6.8.1 Overview

TSensor is the analog temperature detection IP, that is, a digital temperature sensor IP with 8-bit parallel outputs. It detects the temperature of the die and outputs temperature information in binary mode. The resolution is 0.71 ℃/LSB, and the 8-bit data 0–255 corresponds to the temperature range from –40 ℃ to +140 ℃.

IP precision adjustment: Supports linear adjustment of absolute precision. The adjusted precision is ±3 ℃.
The so-called adjustment refers to Trim adjustment in the automatic test equipment (ATE) test. The adjustment precision depends on the test precision.

6.8.2 Features

The TSensor module has the following features:

- Reports the obtained TSensor IP temperature.
  - Assists the algorithm calibration, that is, calculating the average value of 16 temperature samples.
  - Specifies whether to calculate the average value of 16 times for a single point.
  - Reports the calibrated temperature in real time in interrupt mode.
- Supports high and low temperature thresholds.
  - Reports in real time whether the current temperature is higher than the high temperature threshold or lower than the low temperature threshold in interrupt mode.
  - Supports the configuration of high and low temperature thresholds for software.
- Supports ultra-high temperature protection.
  - Reports in real time whether the current temperature exceeds the over-high temperature threshold in interrupt mode.
  - Supports the configuration of the over-high temperature threshold for software.
  - Generates overtemperature protection signals for the RF module.
- Supports over-temperature power-off.
  - Generate power down (PD) signals to disable the logic.
  - Supports the configuration of a temperature threshold for over-temperature power-off for software.
- Reports the sampled temperature periodically.
  - Supports sampling interval for software.
- Supports power gating control.
  - Supports manual control and automatic control.

- Calibrates temperature through ATE.
  - Performs temperature calibration on the TSensor IP by configuring the Trim value.

6.8.3 Operating Modes

The TSensor module can work in the following modes:

- Normal temperature detection mode
- High/Low temperature threshold interrupt mode
- Overtemperature protection interrupt mode
- RF overtemperature protection interrupt mode

NOTE

The preceding modes are enabled or disabled based on the detected temperature. The temperature is determined through periodic sampling and 16 single-point averaging.

6.8.3.1 Normal Temperature Detection Mode

To configure the normal temperature detection mode, perform the following steps:

Step 1 Set GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_sel] to **0x1** and GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_man] to **0x1** to enable the power gating of TSensor in manual mode.

Step 2 Set TSENSOR_AUTO_STS[tsensor_auto_clr] to **0x1** to clear the temperature valid signal generated in automatic mode.

Step 3 Set TSENSOR_TEMP_INT_CLR[tsensor_int_clr] to **0x1** to clear the TSensor interrupt.

Step 4 Set TSENSOR_TEMP_INT_EN[tsensor_done_int_en] to **0x1** to enable the TSensor temperature measurement completion interrupt.

Step 5 Set TSENSOR_CTRL[tsensor_mode] to **0x0** to select the mode of reporting the average value of 16 collected values.

Step 6 Set TSENSOR_CTRL[tsensor_enable] to **0x1** to enable TSensor.

Step 7 Configure TSENSOR_AUTO_REFRESH_PERIOD[tsensor_auto_refresh_period] to set a proper interval for periodic sampling.

Step 8 Set TSENSOR_AUTO_REFRESH_CFG[tsensor_auto_refresh_enable] to **0x1** to enable periodic sampling.

Step 9 Read TSENSOR_TEMP_INT_STS[tsensor_done_int_sts], if the value is **0x1**, wait for the generation of the interrupt signal in 16-sample averaging mode.

Step 10 Read TSENSOR_AUTO_STS[tsensor_data_auto] to obtain the temperature reported in 16-sample averaging single reporting mode.

----End

6.8.3.2 High/Low Temperature Threshold Interrupt Mode

To configure the high/low temperature threshold interrupt mode, perform the following steps:

Step 1 Set GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_sel] to **0x1** and GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_man] to **0x1** to enable the power gating of TSensor in manual mode.

Step 2 Set TSENSOR_AUTO_STS[tsensor_auto_clr] to **0x1** to clear the RDY signal generated in automatic mode.

Step 3 Set TSENSOR_TEMP_INT_CLR[tsensor_int_clr] to **0x1** to clear the TSensor interrupt.

Step 4 Set TSENSOR_TEMP_INT_EN[tsensor_out_thresh_int_en] to **0x1** to enable the TSensor threshold-exceeding interrupt.

Step 5 Configure TSENSOR_TEMP_LIMIT1[tsensor_temp_high_limit] and TSENSOR_TEMP_LIMIT2[tsensor_temp_low_limit] to set proper upper and lower temperature thresholds.

Step 6 Set TSENSOR_CTRL[tsensor_mode] to **0x0** to select the mode of reporting the average value of 16 collected values.

Step 7 Set TSENSOR_CTRL[tsensor_enable] to **0x1** to enable TSensor.

Step 8 Configure TSENSOR_AUTO_REFRESH_PERIOD[tsensor_auto_refresh_period] to set a proper interval for periodic sampling.

Step 9 Set TSENSOR_AUTO_REFRESH_CFG[tsensor_auto_refresh_enable] to **0x1** to enable periodic sampling.

Step 10 Read TSENSOR_TEMP_INT_STS[tsensor_out_thresh_int_sts]. If the value is **1**, the current temperature is higher than the high temperature threshold or lower than the low temperature threshold.

----End

6.8.3.3 Overtemperature Protection Interrupt Mode

To configure the overtemperature protection interrupt mode, perform the following steps:

Step 1 Set GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_sel] to **0x1** and GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_man] to **0x1** to enable the power gating of TSensor in manual mode.

Step 2 Set TSENSOR_AUTO_STS[tsensor_auto_clr] to **0x1** to clear the RDY signal generated in automatic mode.

Step 3 Set TSENSOR_TEMP_INT_CLR[tsensor_int_clr] to **0x1** to clear the TSensor interrupt.

Step 4 Set TSENSOR_TEMP_INT_EN[tsensor_overtemp_int_en] to **0x1** to enable the TSensor overtemperature interrupt.

Step 5 Configure TSENSOR_OVER_TEMP[tsensor_overtemp_thresh] to set a proper overtemperature threshold.

Step 6 Set TSENSOR_OVER_TEMP[tsensor_overtemp_thresh_en] to **0x1** to enable the overtemperature protection.

Step 7 Set TSENSOR_CTRL[tsensor_mode] to **0x0** to select the mode of reporting the average value of 16 collected values.

Step 8 Set TSENSOR_CTRL[tsensor_enable] to **0x1** to enable TSensor.

Step 9 Configure TSENSOR_AUTO_REFRESH_PERIOD[tsensor_auto_refresh_period] to set a proper interval for periodic sampling.

Step 10 Set TSENSOR_AUTO_REFRESH_CFG[tsensor_auto_refresh_enable] to **0x1** to enable periodic sampling.

Step 11 Read TSENSOR_TEMP_INT_STS[tsensor_overtemp_int_sts]. If the value is **1**, the current temperature exceeds the configured overtemperature threshold.

----End

6.8.3.4 RF Overtemperature Protection Interrupt Mode

To configure the RF overtemperature protection interrupt mode, perform the following steps:

Step 1 Set GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_sel] to **0x1** and GATE_TSENSOR_VDDIO[gate_tsensor_vbat_cldo_man] to **0x1** to enable the power gating of TSensor in manual mode.

Step 2 Set TSENSOR_AUTO_STS[tsensor_auto_clr] to **0x1** to clear the RDY signal generated in automatic mode.

Step 3 Set RF_OVER_TEMP_INT_CLR[rf_over_temp_int_clr] to **0x1** to clear the RF overtemperature valid interrupt.

Step 4 Set RF_OVER_TEMP_INT_EN[rf_over_temp_int_en] to **0x1** to enable the RF overtemperature interrupt.

Step 5 Configure TSENSOR_OVER_TEMP[tsensor_overtemp_thresh] to set a proper overtemperature threshold.

Step 6 Set TSENSOR_OVER_TEMP[tsensor_overtemp_thresh_en] to **0x1** to enable the overtemperature protection.

Step 7 Set TSENSOR_CTRL[tsensor_mode] to **0x0** to select the mode of reporting the average value of 16 collected values.

Step 8 Set TSENSOR_CTRL[tsensor_enable] to **0x1** to enable TSensor.

Step 9 Configure TSENSOR_AUTO_REFRESH_PERIOD[tsensor_auto_refresh_period] to set a proper interval for periodic sampling.

Step 10 Set TSENSOR_AUTO_REFRESH_CFG[tsensor_auto_refresh_enable] to **0x1** to enable periodic sampling.

Step 11 Read RF_OVER_TEMP_INT_STS[rf_over_temp_int]. If the value is **1**, the current temperature exceeds the configured overtemperature threshold and is transmitted to the RF module as a valid signal.

----End

6.8.4 Register Summary

Table 6-18 describes the TSensor registers.

Table 6-18 Summary of TSensor registers (base address: 0x4002_8000)

Offset Address	Register	Description	Page
0x0490	RF_TEMP_MODE	RF TEMP mode configuration register	6-111
0x04B4	RF_TEMP_STS	RF TEMP overtemperature status register	6-112
0x04C0	RF_OVER_TEMP_INT_EN	RF TEMP overtemperature interrupt enable register	6-112
0x04C4	RF_OVER_TEMP_INT_CLR	RF TEMP overtemperature interrupt clear register	6-112
0x04C8	RF_OVER_TEMP_INT_STS	RF TEMP overtemperature interrupt register	6-113
0x0500	TSENSOR_START	TSensor start register	6-113
0x0504	TSENSOR_CTRL	TSensor control register	6-113
0x0508	TSENSOR_MAN_STS	TSensor manual control status register	6-114
0x050C	TSENSOR_AUTO_STS	TSensor automatic control status register	6-115
0x0510	TSENSOR_CTRL1	TSensor control register 1	6-115
0x0514	TSENSOR_TEMP_LIMIT1	TSensor temperature upper threshold register	6-116
0x0518	TSENSOR_TEMP_LIMIT2	TSensor temperature lower threshold register	6-117
0x051C	TSENSOR_OVER_TEMP	TSensor overtemperature control register	6-117
0x0520	TSENSOR_TEMP_INT_EN	TSensor interrupt enable register	6-118
0x0524	TSENSOR_TEMP_INT_CLR	RF TEMP configuration register	6-118
0x0528	TSENSOR_TEMP_INT_STS	RF TEMP configuration register	6-119
0x0530	TSENSOR_OVER_TEMP_PD	TSensor overtemperature power-off control register	6-119
0x0540	TSENSOR_AUTO_REFRESH_PERIOD	TSensor automatic detection period configuration register	6-120
0x0544	TSENSOR_AUTO_REFRESH_CFG	TSensor automatic detection enable control register	6-120

6.8.5 Register Description

RF_TEMP_MODE

RF_TEMP_MODE is the RF TEMP mode configuration register.

Offset Address: 0x0490 Total Reset Value: 0xFF44

Bit	Access	Name	Description	Reset
[15:7]	RW	reserved	Reserved	0x1FE
[6]	RW	rf_over_temp_prt_hw_src	Hardware report source select 0: TSensor inside the RF module reports temperature protection alarm. 1: TSensor reports the temperature protection alarm. **Note: Temperature protection alarms can only be reported by TSensor.**	0x1
[5]	-	reserved	Reserved	0x0
[4]	RW	rf_over_temp_prt_polar	Polarity of the overtemperature protection signal 0: Only when the overtemperature protection signal generated by the logic is **0**, the signal is valid. If the signal is **1**, the signal is invalid. 1: Only when the overtemperature protection signal generated by the logic is **1**, the signal is valid. If the signal is **0**, the signal is invalid.	0x0
[3]	RW	rf_over_temp_prt_sel	Overtemperature protection signal select 0: hardware control 1: manual control	0x0
[2]	RW	rf_over_temp_prt_man	Overtemperature protection manual control signal 0: valid 1: invalid	0x1
[1:0]	-	reserved	Reserved	0x0

RF_TEMP_STS

RF_TEMP_STS is the RF TEMP overtemperature status register.

Offset Address: 0x04B4 Total Reset Value: 0x000E

Bit	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x000
[3]	RO	tsensor_overtemp_prt	TSensor overtemperature protection control 0: disabled 1: normal working mode	0x1
[2]	-	reserved	The bit is reserved.	0x1
[1]	RO	rf_over_temp_prt	Overtemperature protection control signal (detected temperature > threshold) 0: disabled 1: normal working mode	0x1
[0]	-	reserved	Reserved	0x0

RF_OVER_TEMP_INT_EN

RF_OVER_TEMP_INT_EN is the RF TEMP overtemperature interrupt enable register.

Offset Address: 0x04C0 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:1]	-	reserved	Reserved	0x0000
[0]	RW	rf_over_temp_int_en	Overtemperature interrupt enable 0: disabled 1: enabled	0x0

RF_OVER_TEMP_INT_CLR

RF_OVER_TEMP_INT_CLR is the RF TEMP overtemperature interrupt clear register.

Offset Address: 0x04C4 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:1]	-	reserved	Reserved	0x0000
[0]	W1_PULSE	rf_over_temp_int_clr	RF overtemperature protection interrupt clear signal 0: invalid 1: interrupt cleared	0x0

RF_OVER_TEMP_INT_STS

RF_OVER_TEMP_INT_STS is the RF TEMP overtemperature interrupt register.

Offset Address: 0x04C8 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:1]	-	reserved	Reserved	0x0000
[0]	RO	rf_over_temp_int	RF overtemperature protection interrupt status 0: invalid 1: valid	0x0

TSENSOR_START

TSENSOR_START is the TSensor start register.

Offset Address: 0x0500 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:1]	-	reserved	Reserved	0x0000
[0]	W1_PULSE	tsensor_start	Temperature code update in automatic mode 0: invalid 1: The temperature code is refreshed once in automatic mode, and the current temperature is obtained by reading TSENSOR_AUTO_STS[tsensor_data_auto]. If TSENSOR_AUTO_STS[tsensor_data_auto] is **1**, the temperature is valid.	0x0

TSENSOR_CTRL

TSENSOR_CTRL is the TSensor control register.

Offset Address: 0x0504 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:4]	-	reserved	Reserved	0x000
[3]	RW	gate_tsensor_vddio_polar	TSensor power gating polarity 0: The TSensor IP is enabled at **1** and disabled at **0**. 1: The TSensor IP is enabled at **0** and disabled at **1**.	0x0
[2:1]	RW	tsensor_mode	TSensor temperature reporting mode 00: 16-sample averaging single reporting	0x0

Bit	Access	Name	Description	Reset
			mode 01: 16-sample averaging cyclic reporting mode 10/11: Single-sample cyclic reporting mode (In this mode, only the temperature code is reported, but is not compared with the threshold.)	
[0]	RW	tsensor_enable	TSENSOR_CTRL switch 0: disabled 1: enabled	0x0

TSENSOR_MAN_STS

TSENSOR_MAN_STS is the TSensor manual control status register.

Offset Address: 0x0508 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:10]	-	reserved	Reserved	0x00
[9:2]	RO	tsensor_data_man	A valid temperature code word of a single point is reported cyclically. The code word **TEMP_OUT** indicating a temperature value has 8 bits, which can be converted into a decimal value ranging from 0 to 255, corresponding to the temperature ranging from –40 ℃ to +140 ℃. The temperature code output is linearly distributed. That is, –40 ℃ corresponds to 0, and +140 ℃ corresponds to 255. Temperature code conversion formula: T (℃) = DEC (TEMP_OUT[7:0]) x 0.705 + (–40) ℃	0x00
[1]	RO	tsensor_rdy_man	Temperature valid signal in single-point cyclic report mode 0: The detection is not started or is being performed manually. 1: The value of **[tsensor_data_man]** is valid.	0x0
[0]	W1_PULSE	tsensor_man_clr	Clears the status of the single-point cyclic report mode. 0: invalid 1: cleared	0x0

TSENSOR_AUTO_STS

TSENSOR_AUTO_STS is the TSensor automatic control status register.

Offset Address: 0x050C Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:10]	-	reserved	Reserved	0x00
[9:2]	RO	tsensor_data_auto	Valid temperature code word in 16-sample averaging single reporting mode or 16-sample averaging cyclic reporting mode. The code word **TEMP_OUT** indicating a temperature value has 8 bits, which can be converted into a decimal value ranging from 0 to 255, corresponding to the temperature ranging from –40 ℃ to +140 ℃. The temperature code output is linearly distributed. That is, –40 ℃ corresponds to 0, and +140 ℃ corresponds to 255. Temperature code conversion formula: T (℃) = DEC (TEMP_OUT[7:0]) x 0.705 + (–40) ℃	0x00
[1]	RO	tsensor_rdy_auto	Temperature valid signal in 16-sample averaging single reporting mode or 16-sample averaging cyclic reporting mode. 0: The automatic detection is not started or is in progress. 1: The value of **[tsensor_data_auto]** is valid.	0x0
[0]	W1_PULSE	tsensor_auto_clr	Clears the status of the 16-sample averaging single reporting mode or the 16-sample averaging cyclic reporting mode. 0: invalid 1: cleared	0x0

TSENSOR_CTRL1

TSENSOR_CTRL1 is the TSensor control register 1.

Offset Address: 0x0510 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:5]	-	reserved	Reserved	0x000
[4]	RW	tsensor_temp_trim_sel	TSensor IP Trim adjustment function source select 0: The temp_trim of the TSensor IP is directly loaded by the eFUSE. 1: The temp_trim of the TSensor IP is selected	0x0

Bit	Access	Name	Description	Reset
			by configuring the register.	
[3:0]	RW	tsensor_temp_trim	Trim value for calibrating the TSensor IP temperature The value is provided based on the ATE test result. Each value corresponds a temperature adjustment value as follows: 0x0: 0.000 ℃ 0x1: 1.410℃ 0x2: 2.820℃ 0x3: 4.230℃ 0x4: 5.640℃ 0x5: 7.050℃ 0x6: 8.460℃ 0x7: 9.870℃ 0x8: 0.000℃ 0X9: –1.410℃ 0xA: –2.820℃ 0xB: –4.230℃ 0xC: –5.640℃ 0xD: –7.050℃ 0xE: –8.460℃ 0xF: –9.870 ℃	0x0

TSENSOR_TEMP_LIMIT1

TSENSOR_TEMP_LIMIT1 is the TSensor temperature upper threshold register.

Offset Address: 0x0514 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x00
[7:0]	RW	tsensor_temp_high_limit	High temperature threshold in 16-sample averaging single reporting mode or the 16-sample averaging cyclic reporting mode. The code word **TEMP_OUT** indicating a temperature value has 8 bits, which can be converted into a decimal value ranging from 0 to 255, corresponding to the temperature ranging from –40 ℃ to +140 ℃. The temperature code output is linearly distributed. That is, –40 ℃ corresponds to 0, and +140 ℃ corresponds to 255. Temperature code conversion formula: T (℃)	0x00

Bit	Access	Name	Description	Reset
			= DEC (TEMP_OUT[7:0]) x 0.705 + (–40) ℃	

TSENSOR_TEMP_LIMIT2

TSENSOR_TEMP_LIMIT2 is the TSensor temperature lower threshold register.

Offset Address: 0x0518 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:8]	-	reserved	Reserved	0x00
[7:0]	RW	tsensor_temp_low_limit	Low temperature threshold in 16-sample averaging single reporting mode or16-sample averaging cyclic reporting mode. The code word **TEMP_OUT** indicating a temperature value has 8 bits, which can be converted into a decimal value ranging from 0 to 255, corresponding to the temperature ranging from –40 ℃ to +140 ℃. The temperature code output is linearly distributed. That is, –40 ℃ corresponds to 0, and +140 ℃ corresponds to 255. Temperature code conversion formula: T (℃) = DEC (TEMP_OUT[7:0]) x 0.705 + (–40) ℃	0x00

TSENSOR_OVER_TEMP

TSENSOR_OVER_TEMP is the TSensor overtemperature control register.

Offset Address: 0x051C Total Reset Value: 0x00FF

Bit	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	RW	tsensor_overtemp_thresh_en	Overtemperature PA protection enable in 16-sample averaging single reporting mode or 16-sample averaging cyclic reporting mode. 0: disabled 1: enabled	0x0
[9:8]	-	reserved	Reserved	0x0
[7:0]	RW	tsensor_overtemp_thresh	Overtemperature PA protection threshold in 16-sample averaging single reporting mode or 16-sample averaging cyclic reporting mode. The code word **TEMP_OUT** indicating a temperature value has 8 bits, which can be	0xFF

Bit	Access	Name	Description	Reset
			converted into a decimal value ranging from 0 to 255, corresponding to the temperature ranging from –40 ℃ to +140 ℃. The temperature code output is linearly distributed. That is, –40 ℃ corresponds to 0, and +140 ℃ corresponds to 255. Temperature code conversion formula: T (℃) = DEC (TEMP_OUT[7:0]) x 0.705 + (–40) ℃	

TSENSOR_TEMP_INT_EN

TSENSOR_TEMP_INT_EN is the TSensor interrupt enable register.

Offset Address: 0x0520 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:3]	-	reserved	Reserved	0x0000
[2]	RW	tsensor_overtemp_int_en	TSensor overtemperature interrupt enable 0: disabled 1: enabled	0x0
[1]	RW	tsensor_out_thresh_int_en	TSensor temperature threshold exceeding interrupt enable 0: disabled 1: enabled	0x0
[0]	RW	tsensor_done_int_en	TSensor temperature measurement completion interrupt enable 0: disabled 1: enabled	0x0

TSENSOR_TEMP_INT_CLR

TSENSOR_TEMP_INT_CLR is the RF TEMP configuration register.

Offset Address: 0x0524 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:1]	-	reserved	Reserved	0x0000
[0]	W1_PULSE	tsensor_int_clr	TSensor interrupt clear 0: invalid 1:cleared	0x0

TSENSOR_TEMP_INT_STS

TSENSOR_TEMP_INT_STS is the RF TEMP configuration register.

Offset Address: 0x0528 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:3]	-	reserved	Reserved	0x0000
[2]	RO	tsensor_overtemp_int_sts	TSensor overtemperature interrupt status (standby) 0: invalid 1: valid	0x0
[1]	RO	tsensor_out_thresh_int_sts	TSensor temperature threshold exceeding interrupt status 0: invalid 1: valid	0x0
[0]	RO	tsensor_done_int_sts	TSensor temperature measurement completion interrupt status 0: invalid 1: valid	0x0

TSENSOR_OVER_TEMP_PD

TSENSOR_OVER_TEMP_PD is the TSensor overtemperature power-off control register.

Offset Address: 0x0530 Total Reset Value: 0x00FF

Bit	Access	Name	Description	Reset
[15:11]	-	reserved	Reserved	0x00
[10]	RW	tsensor_overtemp_pd_en	Overtemperature power-off protection enable in 16-sample averaging single reporting mode or 16-sample averaging cyclic reporting mode. 0: disabled 1: enabled	0x0
[9:8]	-	reserved	Reserved	0x0
[7:0]	RW	tsensor_overtemp_pd_thresh	Overtemperature power-off protection threshold in 16-sample averaging single reporting mode or 16-sample averaging cyclic reporting mode. The code word **TEMP_OUT** indicating the temperature value has 8 bits, which can be converted into decimal values ranging from 0 to 255, corresponding to the temperature ranging from –40 ℃ to +140 ℃. The	0xFF

Bit	Access	Name	Description	Reset
			temperature code output is linearly distributed. That is, –40 ℃ corresponds to 0, and +140 ℃ corresponds to 255. Temperature code conversion formula: T (℃) = DEC (TEMP_OUT[7:0]) x 0.705 + (–40) ℃	

TSENSOR_AUTO_REFRESH_PERIOD

TSENSOR_AUTO_REFRESH_PERIOD is the TSensor automatic detection period configuration register.

Offset Address: 0x0540 Total Reset Value: 0xFFFF

Bit	Access	Name	Description	Reset
[15:0]	RW	tsensor_auto_refresh_period	TSensor automatic detection period (number of 32 kHz clock cycles)	0xFFFF

TSENSOR_AUTO_REFRESH_CFG

TSENSOR_AUTO_REFRESH_CFG is the TSensor automatic detection enable control register.

Offset Address: 0x0544 Total Reset Value: 0x0000

Bit	Access	Name	Description	Reset
[15:1]	RW	reserved	Reserved	0x0000
[0]	RW	tsensor_auto_refresh_enable	Periodic detection enable in 16-sample averaging cyclic reporting mode 0: disabled 1: enabled	0x0

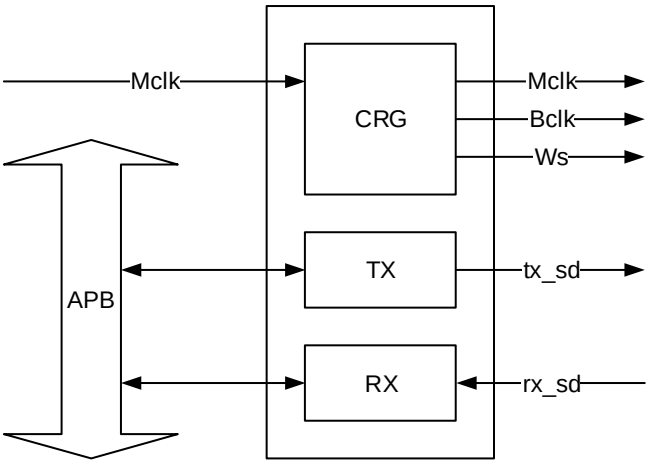
6.9 I²S

6.9.1 Overview

The I²S module works as a slave device on the advanced peripheral bus (APB) while a master device on the I²S bus. The I²S module is used by the CPU to read data from or write data to the slave device on the I²S bus.

The I²S master clock (Mclk) is 12.288 MHz.

Figure 6-18 I²S functional block diagram



6.9.2 Features

The I²S module has the following features:

- Supports 16-bit or 24-bit sampling.
- Supports 8 kHz, 16 kHz, 32 kHz, or 48 kHz sampling rate.

6.9.3 Operating Modes

6.9.3.1 Interface Code

Table 6-19 lists the I²S interface signal.

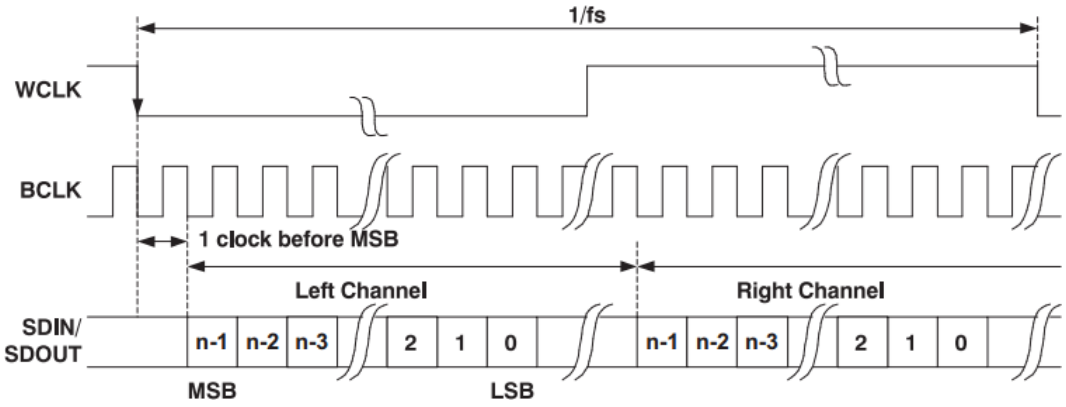
Table 6-19 I²S interface signals

Signal	Width (Bit)	Direction	Description
AUDIO_AIAO_BCLK	1	O	The serial clock (SCLK) or the bit clock (BCLK) refers to each 1-bit data of the digital audio. Each SCLK has one pulse. Serial clock signal (SCK) frequency = 2 x Sampling frequency x Number of sampling bits It is generated and controlled by the master.
AUDIO_AIAO_WS	1	O	Audio-left or audio-right channel select, Word Select (WS) WS is a channel selection signal, indicating the channel selected by the data TX. 0: audio-left channel 1: audio-right channel WS is also called frame clock or left right clock (LRCLK). The WS frequency is equal to the voice sampling rate. It is generated and controlled by the master.
AUDIO_AIAO_MCLK	1	O	Master clock (MCLK) In the ADC/DAC system with the I²S interface, besides the SCK and WS, the coder/decoder (CODEC) requires MCLK provided by the controller. The MCL is generated and controlled by the master, and its frequency is 12.288 MHz.
AUDIO_AIAO_TX_SD	1	O	Output data
AUDIO_AIAO_RX_SD	1	I	Input data

6.9.3.2 Operation Mode

Figure 6-19 shows the I²S operation mode.

Figure 6-19 I²S operation mode



The serial data (SD) is transmitted on the data line as binary complementary codes in the I²S. In the first SCK pulse after the WS change, the most significant bit (MSB) is transmitted first. The MSB of data in the audio-left channel is valid on the second SCK/BCLK rising edge after the WS falling edge, and the MSB of data in the audio-right channel is valid on the second SCK/BCLK rising edge after the WS rising edge.

6.9.3.3 Interrupt Processing

The I²S has the following interrupts (independent interrupt sources, can be disabled, and active high):

- TX_UNFULL_INT

  TX FIFO unfull interrupt. This interrupt is enabled when the number of valid data entries in the TX FIFO is less than 16.

- TX_LEVEL_INT

  TX FIFO interrupt. This interrupt is enabled when the number of valid data entries in the TX FIFO is less than the configured threshold.

- RX_UNEMPTY_INT

  RX FIFO non-empty interrupt. This interrupt is enabled when the number of valid data entries in the RX FIFO is greater than 1.

- RX_LEVEL_INT

  RX FIFO interrupt. This interrupt is enabled when the number of valid data entries in the RX FIFO is greater than the threshold.

- TX_RERROR_INT

  TX FIFO underflow interrupt. This interrupt is enabled when an underflow occurs in the TX FIFO.

- RX_WERROR_INT

  RX FIFO overflow interrupt. This interrupt is enabled when overflow occurs in the RX FIFO.

6.9.3.4 Application Description

Transmitting data in interrupt or query mode is used as an example.

Initialization

The initialization steps are as follows:

Step 1 Set AUDIO_AIAO_CTRL[rx_en] and AUDIO_AIAO_CTRL[tx_en] to **0** to disable the I²S.

Step 2 Configure AUDIO_AIAO_CTRL to set the sampling precision, FIFO threshold, sampling frequency clock (FSCLK) frequency divider, and BCLK frequency divider.

Step 3 In interrupt mode, configure AUDIO_AIAO_CTRL in interrupt mode to enable the corresponding interrupts. In query mode, disable the generation of interrupts.

----End

Transmitting Data

To transmit fixed-length data (for example, four data entries), perform the following steps:

Step 1 Set AUDIO_AIAO_CTRL[tx_en] to **1** to enable the I²S.

Step 4 Read AUDIO_AIAO_INT to check the FIFO threshold status.

- If the value is **1**, the TX FIFO does not reach the threshold. Write four data entries to AUDIO_AIAO_DMA_WR.
- If the value is **0**, the TX FIFO reaches the threshold. Wait until the data to be transmitted arrives the FIFO. If the FIFO does not reach the threshold, data input is allowed.

Step 5 Repeat step Step 4 until all data entries are transmitted.

Step 6 Set AUDIO_AIAO_CTRL[tx_en] to **0** to disable the I²S.

----End

Receiving Data

To receive fixed-length data, perform the following steps:

Step 1 Set AUDIO_AIAO_CTRL[rx_en] to **1** to enable the I²S.

Step 7 Read AUDIO_AIAO_INT to check the RX FIFO threshold status. If the RX FIFO reaches the threshold, read AUDIO_AIAO_DMA_RD and receive four data entries. If the RX FIFO does not reach the threshold, check whether the RX FIFO is empty. If yes, receive one data entry. Otherwise, go to the next step.

Step 8 Repeat step Step 7 until all data entries are received.

Step 9 Set AUDIO_AIAO_CTRL[rx_en] to **0** to disable the I²S.

----End

6.9.3.5 Performance Profiling

16-Bit Sampling

The size of the FIFO in the I²S is 16 x 32 bits. In 16-bit sampling mode, the FIFO can buffer the data of up to 16 sampling points in audio-left and audio-right channels.

When the sampling rate is 8 kHz, the audio data with the maximum duration of 2 ms can be buffered.

When the sampling rate is 16 kHz, the audio data with the maximum duration of 1 ms can be buffered.

When the sampling rate is 32 kHz, the audio data with the maximum duration of 0.5 ms can be buffered.

When the sampling rate is 48 kHz, the audio data with the maximum duration of 0.33 ms can be buffered.

24-Bit Sampling

The size of the FIFO in the I²S is 16 x 32 bits. In 24-bit sampling mode, the FIFO can buffer the data of up to 8 sampling points in the audio-left and audio-right channels.

When the sampling rate is 8 kHz, the audio data with the maximum duration of 1 ms can be buffered.

When the sampling rate is 16 kHz, the audio data with the maximum duration of 0.5 ms can be buffered.

When the sampling rate is 32 kHz, the audio data with the maximum duration of 0.25 ms can be buffered.

When the sampling rate is 48 kHz, the audio data with the maximum duration of 0.16 ms can be buffered.

6.9.4 Register Summary

Table 6-20 describes the I²S registers.

Table 6-20 Summary of I²S registers (base address: 0xF8CC_0000)

Offset Address	Register	Description	Page
0x0000	AUDIO_AIAO_CTRL	Control register	6-126
0x0004	AUDIO_AIAO_RX_FIFO_INT_CLR	RX_FIFO interrupt clear register	6-128
0x0008	AUDIO_AIAO_TX_FIFO_INT_CLR	ATX_FIFO interrupt clear register	6-129
0x000C	AUDIO_AIAO_BCLK_CNT	BCLK_CNT register	6-129
0x0010	AUDIO_AIAO_INT	Interrupt register	6-129
0x0014	AUDIO_AIAO_DMA_WR	DMA write address register	6-131

Offset Address	Register	Description	Page
0x0018	AUDIO_AIAO_DMA_RD	DMA read address register	6-131

6.9.5 Register Description

AUDIO_AIAO_CTRL

AUDIO_AIAO_CTRL is the control register.

Offset Address: 0x0000 Total Reset Value: 0x0070_9000

Bit	Access	Name	Description	Reset
[31:26]	-	reserved	Reserved	0x00
[25]	RW	tx_rerror_int_mask	TX FIFO underflow interrupt enable 0: disabled 1: enabled	0x0
[24]	RW	rx_werror_int_mask	RX FIFO overflow interrupt enable 0: disabled 1: enabled	0x0
[23:21]	RW	aiao_fsclk_div	Frequency ratio of BCLK to WS BCLK = 2 x WS x Number of sampling bits The I²S supports 16-bit and 24-bit sampling. By default, the frequency ratio of BCLK to WS is 64. WS = MCLK/(FSCLK_DIV x BCLK_DIV) 000: 16 001: 32 010: 48 011: 64 100: 128 101: 256 Other values: 8	0x3
[20:17]	RW	aiao_bclk_div	Frequency ratio of MCLK to BCLK 0x0: 1 0x1: 3 0x2: 2 0x3: 4 0x4: 6 0x5: 8	0x8

Bit	Access	Name	Description	Reset
			0x6: 12 0x7: 16 0x8: 24 0x9: 32 0xA: 48 0xB: 64 0xC: 96 Other values: reserved	
[16:14]	RW	dma_tx_level	TX FIFO threshold 000: 2 001: 4 010: 8 011: 12 100: 14 Other values: reserved	0x2
[13:11]	RW	dma_rx_level	RX FIFO threshold 000: 2 001: 4 010: 8 011: 12 100: 14 Other values: reserved	0x2
[10]	RW	aiao_ck_gt_en	Low power consumption enable 0: Clock disabled. The AIAO has no internal clock. 1: Clock enabled. The AIAO internal clock is generated normally.	0x0
[9]	RW	tx_level_int_mask	TX threshold interrupt enable 0: disabled 1: enabled	0x0
[8]	RW	tx_unfull_int_mask	TX unfull interrupt enable 0: disabled 1: enabled	0x0
[7]	RW	rx_level_int_mask	RX threshold interrupt enable 0: disabled 1: enabled	0x0
[6]	RW	rx_unempty_int_mask	RX non-empty interrupt enable 0: disabled	0x0

Bit	Access	Name	Description	Reset
			1: enabled	
[5]	RW	dma_tx_en	DMA TX enable 0: disabled 1: enabled	0x0
[4]	RW	dma_rx_en	DMA RX enable 0: disabled 1: enabled	0x0
[3]	RW	bclk_inv_en	Output BCLK phase reverse enable 0: The output BCLK phase does not change. 1: The Output BCLK phase is inverted. According to the protocol, data transmission can be synchronous with the rising edge or falling edge of the SCK, but the receiving device samples data on the rising edge of the SCK. If the peripheral samples and receives data on the falling edge of the SCK, set [bclk_inv_en] to **1** to make the peripheral BCLK phase and the I2S BCLK phase reverse.	0x0
[2]	RW	tx_en	TX enable 0: disabled 1: enabled	0x0
[1]	RW	rx_en	RX enable 0: disabled 1: enabled	0x0
[0]	RW	data_bit	Bit width 0: 16-bit 1: 24-bit	0x0

AUDIO_AIAO_RX_FIFO_INT_CLR

AUDIO_AIAO_RX_FIFO_INT_CLR is the RX_FIFO interrupt clear register.

Offset Address: 0x0004 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	rx_fifo_int_clr	RX FIFO overflow interrupt clear 0: not cleared 1: cleared	0x0

AUDIO_AIAO_TX_FIFO_INT_CLR

AUDIO_AIAO_TX_FIFO_INT_CLR is the ATX_FIFO interrupt clear register.

Offset Address: 0x0008 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	RW	tx_fifo_int_clr	TX FIFO underflow interrupt clear 0: not cleared 1: cleared	0x0

AUDIO_AIAO_BCLK_CNT

AUDIO_AIAO_BCLK_CNT is the BCLK_CNT register.

Offset Address: 0x000C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:0]	RO	bclk_cnt	Debug register, used to check whether the BCLK is normal	0x00000000

AUDIO_AIAO_INT

AUDIO_AIAO_INT is the interrupt register.

Offset Address: 0x0010 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:12]	-	reserved	Reserved	0x00000
[11]	RO	tx_rerror_int	TX FIFO overflow interrupt source flag (with mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[10]	RO	tx_rerror_int_unmask	TX FIFO overflow interrupt source flag (without mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[9]	RO	tx_level_int	TX FIFO threshold interrupt source flag (with mask) 0: No interrupt is generated.	0x0

Bit	Access	Name	Description	Reset
			1: The interrupt is generated.	
[8]	RO	tx_level_int_unmask	TX FIFO threshold interrupt source flag (without mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[7]	RO	tx_unfull_int	TX FIFO unfull interrupt source flag (with mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[6]	RO	tx_unfull_int_unmask	TX FIFO unfull interrupt source flag (without mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[5]	RO	rx_werror_int	RX FIFO overflow interrupt source flag (with mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[4]	RO	rx_werror_int_unmask	RX FIFO overflow interrupt source flag (without mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[3]	RO	rx_level_int	RX FIFO threshold interrupt source flag (with mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[2]	RO	rx_level_int_unmask	RX FIFO threshold interrupt source flag (without mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[1]	RO	rx_unempty_int	RX FIFO non-empty interrupt source flag (with mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[0]	RO	rx_unempty_int_unmask	RX FIFO non-empty interrupt source flag (without mask) 0: No interrupt is generated. 1: The interrupt is generated.	0x0

AUDIO_AIAO_DMA_WR

AUDIO_AIAO_DMA_WR is the DMA write address register.

Offset Address: 0x0014 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:0]	RW	dma_wr_addr	The DMA writes data to the AIAO through this address.	0x00000000

AUDIO_AIAO_DMA_RD

AUDIO_AIAO_DMA_RD is the DMA read address register.

Offset Address: 0x0018 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:0]	RO	dma_rd_addr	The DMA reads data from the AIAO through this address.	0x00000000

6.10 SDIO

6.10.1 Overview

The SDIO slave controller integrates the secure digital device interfaces that comply with the industrial standard SDIO2.0, and allows the master controller to access the SoC device through the SDIO bus protocol. The host can directly access registers of SDIO interfaces and can access the chip memory by using the DMA. No processor is required.

6.10.2 Features

The SDIO has the following features:

- Supports the SDIO2.0 standard and works only in slave mode.
- Supports access to all memories and registers of the chip.
- Supports clock gating.
- Supports up to 50 MHz clock and a maximum rate of 200 Mbit/s in 4-bit mode.
- Supports uplink, downlink, and loopback tests of the SDIO channel.
- Supports the high-speed mode and SDR25 (single date rate) mode.
- Adjusts the SDIO frequency dynamically.
- Powers off the host side in deep sleep, but keeps the SDIO on the device side powered-on.
- Supports the DMA transfer from the SDIO to the internal RAM through the internal DMA of the SDIO.

6.11 DMA

6.11.1 Overview

Direct memory access (DMA) refers to an operating mode in which I/O data is exchanged by the hardware. In this mode, the DMA controller (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 applies to the transfer of data blocks at a high speed. After receiving a DMA request, the DMAC enables the master bus controller based on the channel settings configured by the CPU, and transmits address and control signals to memories and peripherals. The DMAC also counts the transmitted data entries and reports the data transfer status (that is, whether the data transfer is complete or an error occurs in the data transfer) to the CPU in interrupt mode.

6.11.2 Features

The DMA 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.

  Note: Only master 0 is supported by the chip 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.
- Allows the DMA burst size to be programmed.
- Allows the source address and destination address to be configured 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

  Note: The chip does not support the transfer from a peripheral to another 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.

6.11.3 Operating Modes

To initialize the DMA, perform the following steps:

Step 1 Read DMAC_ENBLD_CHNS to obtain an idle channel ID (**ch_num**).

Step 2 Set DMAC_Cn_CONFIG to **0** to disable the **ch_num** channel and clears the corresponding configuration.

Step 3 To perform DMA data transfer in software request mode, configure DMAC_SOFT_BREQ and DMAC_SOFT_SREQ based on the peripheral IDs in Table 6-21.

Note: Skip Step 3 if the DMA data transfer is performed in hard request mode or the transfer direction is from the memory to the memory.

Table 6-21 DMA interface signals

Peripheral ID	Peripheral Port	Description
0	Uart0_rx	UART0 RX signal
1	Uart0_tx	UART0 TX signal
2	Uart1_rx	UART1 RX signal
3	Uart1_tx	UART1 TX signal
4	Uart2_rx	UART2 RX signal
5	Uart2_tx	UART2 TX signal
6	Spi0_rx	SPI0 RX signal
7	Spi0_tx	SPI0 TX signal
8	Spi1_rx	SPI1 RX signal
9	Spi1_tx	SPI1 TX signal
10	I2s_rx	I^2S RX signal
11	I2s_tx	I^2S TX signal
12–15	-	Reserved

Step 4 Configure the source address DMAC_Cn_SRC_ADDR and destination address DMAC_Cn_DEST_ADDR of the corresponding channel based on the channel ID (**ch_num**).

Step 5 Configure the DMA channel control information (including the transfer length, burst length, and transfer bit width) as required by configuring DMAC_Cn_CONTROL.

Step 6 If transfer with a linked list is required, configure DMAC_CnLLI.

Step 7 Configure DMAC_Cn_CONFIG as required:

- Configure [flow_cntrl] to set the flow control and transfer type.

- Set [dest_peripheral] to a peripheral ID in Table 6-21 to specify the destination device.
- Set [src_peripheral] to a peripheral ID in Table 6-21 to specify the source device.
- Configure [itc] to set the interrupt mask bit.
- Configure [ie] to set the interrupt mask bit.
- Configure [e] to set the channel enable bit.

Step 8 If the interrupt mask bit DMAC_Cn_CONFIG[itc] is not masked, a completion interrupt is reported after DMAC_Cn_CONTROL[transfersize] data entries are transferred, or DMAC_RAW_INT_TC_STATUS is read in polling mode to query the completion status.

----End

6.11.4 Register Summary

Table 6-22 lists the DMAC registers.

Table 6-22 Summary of DMAC registers (base address: 0x4020_0000)

Offset Address	Register	Description	Page
0x000	DMAC_INT_STAT	Interrupt state register	6-135
0x004	DMAC_INT_TC_STAT	DMAC transfer completion interrupt status register	6-136
0x008	DMAC_INT_TC_CLR	Transfer completion status clear register	6-136
0x00C	DMAC_INT_ERR_STAT	Error interrupt status register	6-137
0x010	DMAC_INT_ERR_CLR	Error interrupt clear register	6-137
0x014	DMAC_RAW_INT_TC_STATUS	Raw transfer completion interrupt status register	6-137
0x018	DMAC_RAW_INT_ERR_STATUS	Raw transfer error interrupt status register	6-138
0x01C	DMAC_ENBLD_CHNS	Channel enable register	6-138
0x020	DMAC_SOFT_BREQ	Software burst request register	6-139
0x024	DMAC_SOFT_SREQ	Software single request register	6-139
0x028	DMAC_SOFT_LBREQ	Software last burst request register	6-140
0x02C	DMAC_SOFT_LSREQ	Software last single request register	6-140
0x030	DMAC_CONFIG	DMAC operation configuration register	6-141
0x034	DMAC_SYNC	Synchronization register	6-141
0x100 + n×0x20	DMAC_Cn_SRC_ADDR	DMA source address register	6-142

Offset Address	Register	Description	Page
0x104 + n×0x20	DMAC_Cn_DEST_ADDR	DMA destination address register	6-142
0x108 + n×0x20	DMAC_CnLLI	Linked list register	6-143
0x10C + n×0x20	DMAC_Cn_CONTROL	Channel control register	6-144
0x110 + n×0x20	DMAC_Cn_CONFIG	Channel configuration register	6-148

Table 6-23 describes the value ranges and meanings of the variable in the offset addresses of DMAC registers.

Table 6-23 Variables in the offset addresses of DMAC registers

Variable	Value	Description
n	0–3	DMA channel ID

6.11.5 Register Description

DMAC_INT_STAT

DMAC_INT_STAT is the interrupt status register. Each bit indicates the status of a masked interrupt.

NOTE
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.

Offset address: 0x000 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	RO	int_stat	Masked interrupt status of each DMA channel. bit[0] to bit[3] correspond to DMAC channels 0–3, respectively. The bit values are described as follows: 0: No interrupt is generated. 1: An interrupt is generated. The interrupt request may be an error interrupt or a transfer completion interrupt.	0x0

DMAC_INT_TC_STAT

DMAC_INT_TC_STAT is the DMAC transfer completion interrupt status register. Masked transfer completion interruption status. The corresponding mask bit is DMAC_Cn_CONFIG[itc], where, **n** indicates the channel IDs 0–3.

NOTE
This register must be used with DMAC_INT_STAT to query the interrupt source based on the DMAC_INT_STAT[int_stat] status.

Offset address: 0x004 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	RO	int_tc_stat	Masked transfer completion interrupt status. bit[0] to bit[3] correspond to DMAC channels 0–3, respectively. The bit values are described as follows: 0: No transfer completion interrupt is generated. 1: A transfer completion interrupt is generated.	0x0

DMAC_INT_TC_CLR

DMAC_INT_TC_CLR is the transfer completion clear register. It is used to clear transfer completion interrupts.

Offset address: 0x008 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	WO	int_tc_clr	Clears transfer completion interrupts. bit[0] to bit[3] correspond to DMAC channels 0–3, respectively. The bit values are described as follows: 0: not cleared 1: cleared	0x0

DMAC_INT_ERR_STAT

DMAC_INT_ERR_STAT is the error interrupt status register. It is used to show the masked error interrupt status, and its corresponding mask bit is DMAC_Cn_CONFIG[ie].

Offset address: 0x00C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	RO	int_err_stat	Masked error interrupt status. bit[0] to bit[3] correspond to DMAC channels 0–3, respectively. The bit values are described as follows: 0: No error interrupt is generated. 1: An error interrupt is generated.	0x0

DMAC_INT_ERR_CLR

DMAC_INT_ERR_CLR is the transfer error interrupt clear register. It is used to clear error interrupts.

Offset address: 0x010 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	WO	int_err_clr	Error interrupt clear. bit[0] to bit[3] correspond to DMAC channels 0–3, respectively. The bit values are described as follows: 0: not cleared 1: cleared	0x0

DMAC_RAW_INT_TC_STATUS

DMAC_RAW_INT_TC_STATUS is the raw transfer completion interrupt status register. It is used to show the status of the raw transfer completion interrupts of each channel.

Offset address: 0x014 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	RO	raw_int_tc_stat	Raw transfer completion interrupt status. bit[0] to bit[3] correspond to DMAC channels 0–3, respectively.	0x0

Bit	Access	Name	Description	Reset
			The bit values are described as follows: 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 the raw transfer error interrupt status register. It is used to show the status of the raw error interrupts of each channel.

Offset address: 0x018 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	RO	raw_int_err_stat	Raw error interrupt status of each channel. bit[0] to bit[3] correspond to DMAC channels 0–3, respectively. The bit values are described as follows: 0: No error interrupt is generated. 1: An error interrupt is generated.	0x0

DMAC_ENBLD_CHNS

DMAC_ENBLD_CHNS is the channel enable register.

NOTE
The enable bit in the DMAC_Cn_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: 0x01C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:4]	-	reserved	Reserved	0x0000000
[3:0]	RO	enabled_channels	Channel enable status. bit[0] to bit[3] correspond to DMAC channels 0–3, respectively. The bit values are described as follows: 0: disabled 1: enabled	0x0

DMAC_SOFT_BREQ

DMAC_SOFT_BREQ is the software burst request register. Software controls the generation of 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: 0x020 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	soft_breq	Whether to generate a DMA burst transfer request. For details about the mapping between the bit[0]–bit[15] and interface signals, see Table 6-21. When this register is written: 0: no impact 1: A DMA single transfer request is generated. When the transfer is complete, the corresponding bit is cleared. When this register is read: 0: The peripheral corresponding to the request signal DMACBREQ[15:0] does not send a DMA burst request. 1: The peripheral corresponding to the request signal DMACBREQ[15:0] is requesting a DMA burst transfer.	0x0000

DMAC_SOFT_SREQ

DMAC_SOFT_SREQ is the software single request register. Software controls the generation of 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.

Offset address: 0x024 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	soft_sreq	Whether to generate a DMA single transfer request.	0x0000

Bit	Access	Name	Description	Reset
			When this register is written: 0: not cleared 1: A DMA single transfer request is generated. When the transfer is complete, the corresponding bit is cleared. When this register is read: 0: The peripheral corresponding to the request signal DMACBREQ[15:0] does not send a DMA single request. 1: The peripheral corresponding to the request signal DMACBREQ[15:0] is requesting a single DMA transfer.	

DMAC_SOFT_LBREQ

DMAC_SOFT_LBREQ is the software last burst request register. Software controls the generation of a DMA burst transfer request.

Offset address: 0x028 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	WO	soft_lbreq	Whether to generate a DMA last burst transfer request by the software. 0: not cleared 1: A DMA last burst transfer request is generated. When the transfer is complete, the corresponding bit is cleared.	0x0000

DMAC_SOFT_LSREQ

DMAC_SOFT_LSREQ is the last single transfer request register of the software. Software controls the generation of a DMA burst transfer request.

Offset address: 0x02C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	WO	soft_lsreq	Whether to generate a DMA last single transfer request by the software 0: not cleared 1: A DMA last single transfer request is generated. When the transfer is complete, the	0x0000

Bit	Access	Name	Description	Reset
			corresponding bit is cleared.	

DMAC_CONFIG

DMAC_CONFIG is the DMAC operation configuration register. The endianness of the two master interfaces of the DMAC can be changed by configuring DMAC_CONFIG[m1] and [m2]. After the register is reset, the two master interfaces are set to little endian mode.

NOTE
The two masters use little endian.

Offset address: 0x030 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2]	RW	m2	Endianness mode of master 2 0: little endian 1: big endian	0x0
[1]	RW	m1	Endianness mode of master 1 0: little endian 1: big endian	0x0
[0]	RW	e	DMAC enable 0: disabled 1: enabled	0x0

DMAC_SYNC

DMAC_SYNC is the synchronization register.

Offset address: 0x034 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:16]	-	reserved	Reserved	0x0000
[15:0]	RW	dmac_sync	Sync logic enable for DMA request signals of the corresponding peripheral 0: enabled 1: disabled	0x0000

DMAC_Cn_SRC_ADDR

DMAC_Cn_SRC_ADDR is the DMA source address register, providing source address of the data to be transmitted (ordered by byte).

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 linked list nodes.

When the channel is active, no valid information can be obtained by reading this register. This is because even when the software obtains the value of this register, the value of this register changes with the channel transfer. When the channel stops transferring, reading this register obtains the source address when the DMAC reads the last item.

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

Offset address: 0x100 + n x 0x20 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:0]	RW	src_addr	DMA source address	0x00000000

DMAC_Cn_DEST_ADDR

DMAC_Cn_DEST_ADDR is the destination address register, providing the destination address of the data to be transmitted (sorted by byte).

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 linked list nodes.

When the channel is active, no valid information can be obtained by reading this register. This is because even when the software obtains the value of this register, the value of this register changes with the channel transfer. When the channel stops transferring, reading this register obtains the source address when the DMAC reads the last item.

Offset address: 0x104 + n x 0x20 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:0]	RW	dest_addr	DMA destination address	0x00000000

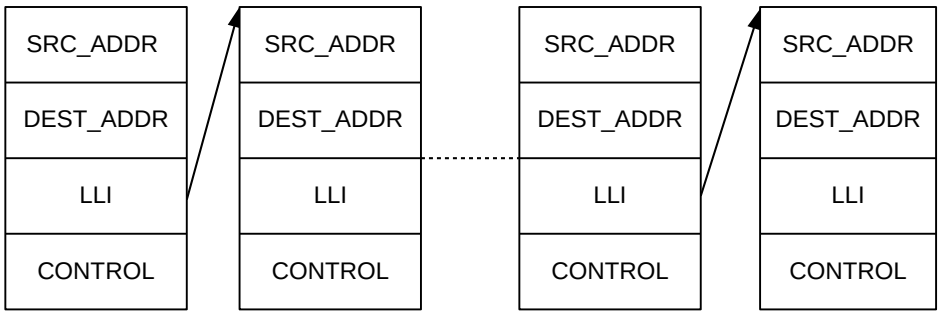
DMAC_CnLLI

DMAC_CnLLI is the linked list register.

The data structure of the DMAC linked list node is as follows:

- DMAC_Cn_SRC_ADDR: sets the start address of the source device.
- DMAC_Cn_DEST_ADDR: sets the start address of the destination device.
- DMAC_CnLLI: sets the next node address.
- DMAC_Cn_CONTROL: sets parameters such as the master, bit width, burst size, address increment, and transfer size for accessing the source and destination devices.

Table 6-24 Structure of the DMAC linked list



NOTICE

DMAC_CnLLI[lli] cannot be set to a value greater than **0xFFFF_FFF0**. Otherwise, a 4-word burst is transferred, and the address is rewound to **0x00000000**. As a result, the data structure of the linked list nodes cannot be stored in consecutive address areas.

If the value of **[lli]** 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 of the current node are transferred.

Offset address: 0x108 + n x 0x20 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:2]	RW	lli	bit[31:2] in the next linked list node address (bit[1:0] of the address is set to **0**.) **Note: The linked list address must be 4-byte aligned.**	0x00000000
[1]	RW	reserved	Reserved **Note: This bit value must be 0 during write operations, and this bit must be masked during read operations.**	0x0
[0]	RW	lm	Master for loading the next linked list node 0: master 1 1: master 2	0x0

DMAC_Cn_CONTROL

DMAC_Cn_CONTROL is the channel control register, containing the DMA channel control information, such as the transfer length, burst length, 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.

When the channel is active, no valid information can be obtained by reading this register. This is because even when the software obtains the value of this register, the value of this register changes with the channel transfer. Read this register when the channel stops transferring.

Offset address: 0x10C + n x 0x20 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31]	RW	i	Whether to trigger the transfer completion interrupt through the current linked list node 0: no 1: yes	0x0
[30:28]	RW	prot	HPROT[2:0] access protection signal transmitted by the master	0x0
[27]	RW	di	Destination address increment 0: not incremented 1: The value increases by one each time a number 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.**	0x0
[26]	RW	si	Source address increment 0: not progressively 1: The value increases by one each time a number is transferred. **If the source device is a peripheral, the source address is not incremented. If the source device is a memory, the source address is incremented.**	0x0
[25]	RW	d	Master select for accessing the destination device 0: master 1 1: master 2	0x0
[24]	RW	s	Master select for accessing the source device 0: master 1 1: master 2	0x0
[23:21]	RW	dwidth	Transfer bit width of the destination device The transfer bit width is invalid if it is greater	0x0

Bit	Access	Name	Description	Reset
			than the bit width of the master. The data widths of the destination and source devices can be different. The hardware automatically packs and unpacks the data. For details about the mapping between the values of DWidth and the bit width, see Table 6-22.	
[20:18]	RW	swidth	Transfer bit width of the source device The transfer bit width is invalid if it is greater than the bit width of the master. The data widths of the destination and source devices can be different. The hardware automatically packs and unpacks the data. For details about the mapping between DWidth/SWidth and the bit width, see Table 6-22.	0x0
[17:15]	RW	dbsize	Burst length of the destination device Indicates the number of data entries to be transferred in a burst transfer of the destination device, that is, number of data entries transferred when DMACCnBREQ is valid. This value must be set to a burst size supported by the destination device. If the destination device is a memory, the value is set to the storage address boundary. For details about the mapping between DBSize/SBSize and the transfer length, see Table 6-21.	0x0
[14:12]	RW	sbsize	Burst length of the source device Indicates the data entries to be transferred in a burst transfer of the source device, that is, number of data entries transferred when DMACCnBREQ is valid. The value must be set to a burst size supported by the source device. If the source device is a memory, the value is set to the storage address boundary. For details about the mapping between SBSize and the transfer length, see Table 6-21.	0x0
[11:0]	RW	transfersize	Write: When the DMAC is a flow controller, this field indicates the number of data entries to be transferred by the source device. Read: Obtains the number of data entries that have been transferred from the bus connected to the destination device. When the channel is active, no valid	0x000

Bit	Access	Name	Description	Reset
			information can be obtained by reading this register. This is because even when the software obtains the value of this register, the value of this register changes with the channel transfer. Therefore, read this register when the channel stops transferring.	

Table 6-25 lists the values of DBSize and SBSize of DMAC_Cn_CONTROL and the corresponding burst length.

Table 6-25 Mapping between DBSize/SBSize and the burst length

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

Table 6-26 lists the value of DWidth and SWidth of DMAC_Cn_CONTROL and the corresponding transfer bit width.

Table 6-26 Mapping between DWidth/SWidth and the transfer bit width

SWidth or DWidth	Transfer Bit Width
000	Byte (8 bits)
001	Halfword (16 bits)
010	Word (32 bits)
011	reserved
100	reserved
101	reserved
110	reserved
111	reserved

Note the following when configuring DMAC_Cn_CONTROL:

- When the transfer bit width of the source device is smaller than that of the destination device, the value of **[transfer bit width x 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_Cn_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 write operations, this field serves as a control register, determining the number of data entries to be transferred by the DMAC. During read operations, this field serves as a status register, returning the number of the remaining data entries to be transferred (unit: bit width of the source device).
- When the transfer size field is set to a value greater than the depth of the FIFO (peripheral FIFO, 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 DMAC_Cn_CONTROL[prot] and DMAC_Cn_CONFIG[1] set by programming the channel registers. Table 6-27 describes the meanings of the three protection bits of the prot field.

Table 6-27 Definitions of the prot field of DMAC_Cn_CONTROL

Bit Field	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 that when this bridge sees the first read of a burst of eight digits, this bridge can transfer the whole burst of eight reads on the destination bus, rather than pass 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].

Bit Field	Description	Purpose
[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_Cn_CONFIG

DMAC_Cn_CONFIG is the channel configuration register.

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

Offset address: 0x110 + n x 0x20 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:19]	-	reserved	Reserved **Note: This bit value must be 0 during write operations, and this bit must be masked during read operations.**	0x0000
[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 works with bits [a] and [e_c] to disable a DMA channel without data loss.	0x0
[17]	RO	a	Active bit. 0: There is no data in the channel FIFO. 1: There is data in the channel FIFO. This bit works with bits [h] and [e_c] to disable a DMA channel without data loss.	0x0

Bit	Access	Name	Description	Reset
[16]	RW	l	Lock bit. 0: disables the lock transfer on the bus. 1: enables the lock transfer on the bus.	0x0
[15]	RW	itc	Channel transfer completion interrupt mask. 0: enabled 1: disabled	0x0
[14]	RW	ie	Channel error interrupt mask. 0: enabled 1: disabled	0x0
[13:11]	RW	flow_cntrl	Flow control 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 about the flow control type and transfer type, see Table 6-28.	0x0
[10]	-	reserved	Reserved **Note: This bit value must be 0 during write operations, and this bit must be masked during read operations.**	0x0
[9:6]	RW	dest_peripheral	Destination device. The field is used to select a peripheral request signal as the request signal for the DMA destination device of the channel. **Note: If the destination device of the DMA transfer is a memory, ignore this value.**	0x0
[5]	-	reserved	Reserved **Note: This bit value must be 0 during write operations, and this bit must be masked during read operations.**	0x0
[4:1]	RW	src_peripheral	Source device. This field is used to select a peripheral request signal as the request signal for the DMA source device of the channel. **Note: If the source device of the DMA transfer is a memory, ignore this value.**	0x0
[0]	RW	e_c	Channel enable. 0: disabled 1: enabled When this bit is cleared (that is, the channel is disabled), the current bus transfer continues until the transfer is complete. Then the	0x0

Bit	Access	Name	Description	Reset
			channel is disabled and the remaining data in the FIFO is lost. When the last linked list item (LLI) is transferred or an error occurs during transfer, the channel is disabled and this bit is cleared. To disable the channel but retain the data in the channel FIFO, [h] must be set to **1** so that the channel ignores subsequent DMA requests. Then [a] must be polled until its value is **0**, indicating that no data is left in the channel FIFO. At this time, [e_c] can be cleared. **Note: To enable a channel by setting this bit to 1, you must reinitialize the channel first. If this operation is not performed before enabling the channel, unexpected errors may occur.**	

NOTICE

When a channel is disabled by setting [e_c], [e_c] can be reset again only after the poling result of the corresponding bit in DMAC_ENBLD_CHNS is **0**. This is because the channel is not disabled immediately after [e_c] is cleared. This also applies to the running delay of the bus burst.

Table 6-28 describes the flow control and transfer types corresponding to the flow_cntrl field of the DMAC_Cn_CONFIG register.

Table 6-28 Definition of flow controllers and transfer types

Bit Value	Transmission type	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

6.12 ADC

6.12.1 Overview

LSADC is a successive approximations register (SAR) ADC that converts analog signals into digital signals.

Table 6-29 ADC Specifications

Parameter	Min.	Typical	Max	Unit	Description
Power supply					
AVDD	1.71	1.8	1.89	V	Voltage of the analog power
DVDD	0.99	1.1	1.21	V	Voltage of the digital power
LSADC					
Full Scale Intput	2.5	-	3.6	V	ADC value range
DNL	-	±1.5	-	LSB	Differential nonlinearity (DNL)
INL	-	±1.5	-	LSB	Integral nonlinearity (INL)
Resolution	-	10	-	bit	Precision
Power Dissipation	-	0.8	-	mA	Operating power consumption
	-	1	-	μA	Power-off power consumption
Clock					
f_{CLK}	-	-	3	MHz	Input clock frequency
DC	45	50	55	%	Duty cycle
f_S	-	-	157	Ksps	Sampling ratio
T_{CONV}	-	15	-	CLK cycle	Data conversion time

6.12.2 Features

The LSADC has the following features:

- Supports input clock of 3 MHz, 12-bit sampling, and single-channel sampling frequency less than 200 kHz.
- Supports eight channels in total. The software can enable any of channels 0–7. The logic initiates channel switching based on the channel ID (that is, channel priority) in ascending order. One sample is collected for each channel. During channel switching, the ADC circuit is reset once.
- The time register from reset deassertion to ADC data conversion start is configurable.
- Supports the 128 x 15-bit FIFO for data cache. The upper 3 bits indicate the channel ID, and the lower 12 bits indicate the valid data.
- Supports average filtering of ADC sampling data. The average times can be 1 (not averaged), 2, 4, or 8. In multi-channel mode, each channel receives N (average number of filtered data entries) data entries and then switches to another channel.
- Reports FIFO threshold interrupts and full interrupts.

NOTICE

Channel 7 of the LSADC is the internal VBAT voltage detection channel, instead of a pin input.

6.12.3 Operating Modes

To configure the ADC, perform the following steps:

Step 1 The CPU sets the scanning channel ID by configuring LSADC_CTRL0[ch_vld] to enable one or more channels. Set the average filtering mode by configuring LSADC_CTRL0[equ_model_sel].

Step 2 Configure LSADC_CTRL1[rxintsize], LSADC_CTRL2[rxim], and LSADC_CTRL2[rorim] to set the FIFO interrupt and threshold.

Step 3 Set LSADC_CTRL11[power_down] to **0** to power on the ADC.

Step 4 Set LSADC_CTRL7[start] to **1** to start ADC sampling.

Step 5 Read LSADC_CTRL9[dr] when an interrupt is reported to obtain ADC data.

Step 6 Set LSADC_CTRL8[stop] to **1** and set LSADC_CTRL11[power_down] to **1** to stop sampling.

Step 7 Read LSADC_CTRL9[dr] to obtain ADC data until the FIFO is empty, that is, LSADC_CTRL10[rne] is **0**.

----End

NOTICE

- LSADC_CTRL7[start] and LSADC_CTRL8[stop] must be configured together. That is, configure [start] when the sampling starts and configure [stop] after the ADC sampling ends. If [stop] is not configured after the first conversion, configuring [start] for the second ADC conversion causes data errors, that is, the start→start→stop operation is not supported.
- Data in the FIFO must be read empty after the current sampling, that is, after [start] is configured. Otherwise, residual data of the last ADC sampling will exist in the next sampling.
- The interval between the time when the first conversion ends ([stop] is configured) and the time when the second AD conversion starts ([start] is configured) is greater than 6 μs (more than 15 clk_3m cycles).

6.12.4 Register Summary

Table 6-30 describes the ADC registers.

Table 6-30 Summary of ADC registers (base address: 0x4007_0000)

Offset Address	Register	Description	Page
0x000	LSADC_CTRL0	ADC control register	6-154
0x004	LSADC_CTRL1	ADC FIFO configuration register	6-154
0x008	LSADC_CTRL2	ADC interrupt control register	6-155
0x00C	LSADC_CTRL3	ADC interrupt clear register	6-155
0x010	LSADC_CTRL4	ADC FIFO status register	6-156
0x014	LSADC_CTRL5	ADC raw interrupt status register	6-156
0x018	LSADC_CTRL6	ADC masked interrupt state register	6-157
0x01C	LSADC_CTRL7	ADC start scanning control register	6-157
0x020	LSADC_CTRL8	ADC stop scanning control register	6-157
0x024	LSADC_CTRL9	ADC FIFO read data control register	6-158
0x028	LSADC_CTRL10	ADC reserve control register	6-158
0x02C	LSADC_CTRL11	ADC power-off control register	6-159

6.12.5 Register Description

LSADC_CTRL0

LSADC_CTRL0 is the ADC control register.

Offset address: 0x000 Total Reset Value: 0x0000_F000

Bit	Access	Name	Description	Reset
[31:26]	RW	reserved	Reserved	0x00
[25:24]	RW	cur_bais	Analog power control 10: manual control (AVDD = 1.8 V). 11: manual control (AVDD = 3.3 V). Other values: auto identification mode	0x0
[23:12]	RW	rst_cnt	Count time from the reset (RST) to the start of the conversion The value must be greater than or equal to 15.	0x00F
[11:10]	RW	reserved	Reserved	0x0
[9:8]	RW	equ_model_sel	Average algorithm mode select 00: 1-time average (that is, not average) 01: 2-time average algorithm 10: 4-time average algorithm 11: 8-time average algorithm	0x0
[7:0]	RW	ch_vld	Whether the channel is valid bit[0]–bit[7] correspond to channels A–H, respectively. The bit values are described as follows: 0: invalid 1: valid	0x00

LSADC_CTRL1

LSADC_CTRL1 is the ADC FIFO configuration register.

Offset address: 0x004 Total Reset Value: 0x0000_0002

Bit	Access	Name	Description	Reset
[31:3]	RW	reserved	Reserved	0x00000000
[2:0]	RW	rxintsize	FIFO threshold setting When the number of data entries is greater than or equal to the configured words, RXRIS is valid.	0x2

Bit	Access	Name	Description	Reset
			000: 127 001: 124 010: 64 011: 32 100: 16 101: 8 110: 4 111: 1	

LSADC_CTRL2

LSADC_CTRL2 is the ADC interrupt control register.

Offset address: 0x008 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:2]	RW	reserved	Reserved	0x00000000
[1]	RW	rxim	Threshold interrupt mask 0: enabled 1: disabled	0x0
[0]	RW	rorim	FIFO overflow interrupt mask 0: enabled 1: disabled	0x0

LSADC_CTRL3

LSADC_CTRL3 is the ADC interrupt clear register.

Offset address: 0x00C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:1]	RW	reserved	Reserved	0x00000000
[0]	W1_PULSE	roric	FIFO overflow interrupt clear 0: not cleared 1: cleared	0x0

LSADC_CTRL4

LSADC_CTRL4 is the ADC FIFO status register.

Offset address: 0x010 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:3]	-	reserved	Reserved	0x00000000
[2]	RO	bsy	ADC busy indicator 0: idle 1: busy	0x0
[1]	RO	rff	Whether the FIFO is full 0: no 1: yes	0x0
[0]	RO	rne	Whether the FIFO is empty 0: yes 1: no	0x0

LSADC_CTRL5

LSADC_CTRL5 is the ADC raw interrupt status register.

Offset address: 0x014 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RO	rxris	Raw RX FIFO interrupt status 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[0]	RO	rorris	Raw RX overflow interrupt 0: No interrupt is generated. 1: The interrupt is generated.	0x0

LSADC_CTRL6

LSADC_CTRL6 is the ADC masked interrupt status register.

Offset address: 0x018 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:2]	-	reserved	Reserved	0x00000000
[1]	RO	rxmis	Masked RX FIFO interrupt status 0: No interrupt is generated. 1: The interrupt is generated.	0x0
[0]	RO	rormis	Masked RX overflow interrupt state 0: No interrupt is generated. 1: The interrupt is generated.	0x0

LSADC_CTRL7

LSADC_CTRL7 is the ADC start scanning control register.

Offset address: 0x01C Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	W1_PULSE	lsadc_start	LSADC start signal. 0: invalid 1: LSADC started	0x0

LSADC_CTRL8

LSADC_CTRL8 is the ADC stop scanning control register.

Offset address: 0x020 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:1]	-	reserved	Reserved	0x00000000
[0]	W1_PULSE	lsadc_stop	Stops auto scanning. 0: invalid 1: disables the scanning function of the LSADC (Note: the scanning function can be restarted only after LSADC_CTRL7[lsadc_start] is written).	0x0

LSADC_CTRL9

LSADC_CTRL9 is the ADC FIFO read data control register.

Offset address: 0x024 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:15]	-	reserved	Reserved	0x00000
[14:0]	RO	dr	Reads ADC data. bit[14:12]: channel ID of the current data (Channel IDs 0x0–0x7 correspond to channels A–H, respectively.) bit[11:0]: valid data (Note, bit[1:0] indicates the decimal place.)	0x0000

LSADC_CTRL10

LSADC_CTRL10 is the ADC reserve control register.

Offset address: 0x028 Total Reset Value: 0x0000_0000

Bit	Access	Name	Description	Reset
[31:28]	RW	reserved	Reserved	0x0
[27:24]	RW	lsadc_reg_3to0	General register 4	0x0
[23:22]	RW	reserved	Reserved	0x0
[21:16]	RW	lsadc_reg2_5to0	General register 3	0x00
[15:14]	RW	reserved	Reserved	0x0
[13:12]	RW	lsadc_reg1_7to6	General register 2	0x0
[11:9]	RW	reserved	Reserved	0x0
[8]	RW	lsadc_reg1_3	General register 1	0x0
[7]	RW	reserved	Reserved	0x0
[6:0]	RW	lsadc_reg0_6to0	General register 0	0x00

LSADC_CTRL11

LSADC_CTRL11 is the ADC power-off control register.

Offset address: 0x02C Total Reset Value: 0x0000_0001

Bit	Access	Name	Description	Reset
[31:1]	RW	reserved	Reserved	0x00000000
[0]	RW	power_down	ADC power-down control 0: normal working 1: powered down	0x1

7 JTAG

7.1 Overview

Hi3861, Hi3861L, and Hi3881 are integrated with a self-developed CPU. These chips provide built-in JTAG debugging interfaces and integrate the Coresight debugging architecture, supporting both the Coresignt-based JTAG and serial wire debug (SWD) interfaces.

By default, the built-in interface of the CPU is used for debugging. To use the Coresight debugging interface, enable register DBG_PORT_SEL[dbg_port_sel] to **1** to switch to the Coresight debugging mode.

Hi3861, Hi3861L, and Hi3881 support the Lauterbach and J-Link emulators. After the emulator is inserted, you need to select a connection mode according to the value of the debugging mode enable register DBG_PORT_SEL. The connection mode can be the default JTAG mode, or Coresignt-based JTAG or SWD modes. Otherwise, the emulator cannot be connected to the CPU.

7.2 Debugging Interface

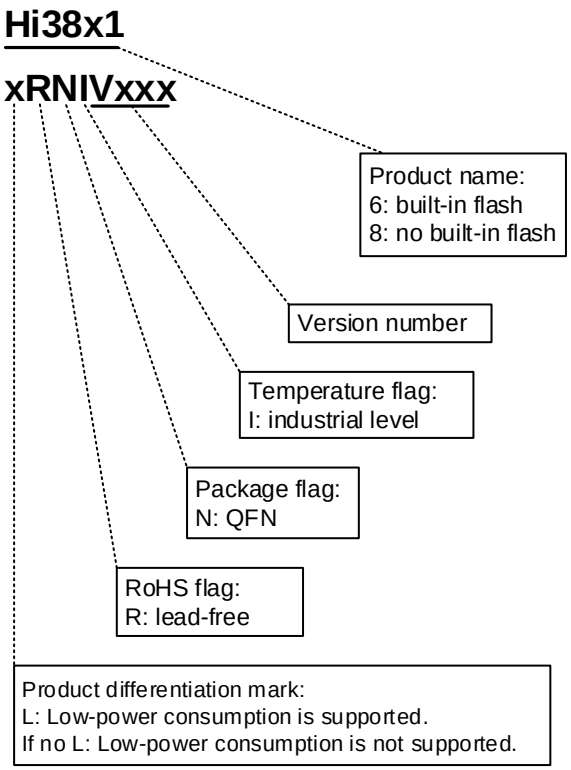
NOTICE

The chip debugging interface is multiplexed as other functions by default. To use the debugging function, set the GPIO_08 pin to high level during power-up to deassert the system reset. Then, the pin is reset as a debugging interface. The other functions of the GPIO_08 pin are not affected and the GPIO_08 pin can be used as a normal pin.

For details about the mapping between debugging pins and PAD names, see the *Hi3861 V100/Hi3861L V100/Hi3881 V100 Wi-Fi Chip Data Sheet (Hardware)*.

A Ordering Information

Figure A-1 Chip mark naming convention



Note: In the version number, xxx indicates the number that can be changed. It is for internal use only.

Table A-1 Configurations of Hi3861, Hi3861L, and Hi3881

Part Number	Packaging Type	Package Dimension	Pitch
Hi3861 RNIV100	QFN 32	5 mm x 5 mm	0.4 mm
Hi3861L RNIV100	QFN 32	5 mm x 5 mm	0.4 mm
Hi3881 RNIV100	QFN 32	5 mm x 5 mm	0.4 mm

B Acronyms and Abbreviations

A

ABB	Analog Baseband
ACK	Acknowledgement
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
AHB	Advanced High Performance Bus
AMBA	Advanced Microcontroller Bus Architecture
A-MPDU	Aggregate MAC Protocol Data Unit
APB	Advanced Peripheral Bus
ASIC	Application-Specific Integrated Circuit
ATE	Automatic Test Equipment
AVDD	Analog Voltage Device Drain

B

BCC	Binary Convolutional Code
BCLK	Bit Clock
BLE	Bluetooth Low Energy
BPSK	Binary Phase Shift Keying
BSS	Basic Service Set
BT	Bluetooth

C

CBB	Common Building Block
CBC	Cipher Block Chaining
CBUS	C-BUS
CCA	Clear Channel Assessment
CCK	Complementary Code Keying
CG	Clock Gating
CMU	Clock Managing Unit
CODEC	Coder/Decoder
CPU	Central Processing Unit
CRG	Clock and Reset Generator
CSI	Channel State Information
CTR	Counter

D

DAC	Digital Analog Converter
DC	Direct Current
DC	Duty Cycle
DBAC	Dual Band Adaptive Concurrent
DBB	Digital Baseband
DBDC	Dual Band Dual Concurrent
DFT	Design For Testability
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
DNL	Differential Nonlinearlty
DPD	Digital Pre-Distortion
DRBG	Deterministic Random Bit Generator
DS	Drive Strength
DSSS	Direct Sequence Spread Spectrum
DTIM	Delivery Traffic Indication Map
DVDD	Digital Voltage Device Drain

E

ECB	Electronic Code Book

ECC	Elliptic Curve Cryptography
EFUSE	Electrical FUSE
EIFS	Extended Inter-Frame Space
EMI	Electromagnetic Interference
F	
FIFO	First In First Out
FIPS	Federal Information Processing Standards
FOUT	Frequency Output
FSCLK	Sampling Frequency Clock
FTM	Fine Timing Measurement
G	
GI	Guard Interval
GPIO	General Purpose Input/Output
H	
HMAC	Hash-based Message Authentication Code
HPI	Hardware Platform Interface
HPM	Hardware Performance Monitor
HT	High Throughput
HVT	High Voltage Threshold
I	
I2C	The Inter-Integrated Circuit
I2S	Inter-IC Sound
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IF	Interface
INL	Integral Nonlinearity
IO	Input Output
IP	Intelligent Property
ISA	Instruction Set Architecture

IV	Initialization Vector
J	
JTAG	Joint Test Action Group
K	
KDF	Key Derivation Function
L	
LDO	Low Dropout Regulator
LNA	Low Noise Amplifier
LO	Local Oscillator
LPF	Low Pass Filter
LRCLK	Left Right Clock
LSADC	Low-Speed Analog-to-Digital Converter
LSB	Least Significant Bit
LUT	Lookup Table
LVT	Low Voltage Threshold
M	
MAC	Media Access Control
MCLK	Master Clock
MCU	Main Control Unit
MSB	Most Significant Bit
MU-MIMO	Multi-User Multiple-Input Multiple-Output
MUX	Multiplexer
N	
NMI	Non-Maskable Interrupt
O	

OFDM	Orthogonal Frequency Division Multiplex
OSC	Oscillator
OTT	Over The Top
OVP	Overvoltage Protection

P

P2P	Peer-to-Peer
PA	Power Amplifier
PD	Power Down
PHY	Physical Layer
PKE	Public-Key-Engine
PLL	Phase-Locked Loop
PMP	Physical Memory Protection
PMU	Power Management Unit
POR	Power On Reset
PP	Page Program
PPM	Parts Per Million
PSM	Power Saving Mode
PTA	Packet Traffic Arbitration
PWM	Pulse-Width Modulation

Q

QAM	Quadrature Amplitude Modulation
QFN	Quad Flat Non-leaded package
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying

R

RAM	Random Access Memory
RDSR	Read Status Register
REF	Reference Indicator
RF	Radio Frequency
RIFS	Reduced Interface Space
RISC	Reduced Instruction Set Computing
RO	Ring Oscillator
ROM	Read-Only Memory
RSA	Rivest-Shamir-Adleman
RSSI	Received Signal Strength Indicator
RTC	Real-Time Clock
RX	Receiver

S

SAR	Successive Approximations Register
SCK	Serial Clock singal
SCL	Serial Clock Line
SCLK	Serial Clock
SCO	Sampling Clock Offset
SD	Serial Data
SD	Secure Digital
SDA	Serial Data and Address
SDIO	Secure Digital Input/Output
SDR	Single Date Rate

SFC	Serial Peripheral Interface Flash Controller
SHA	Secure Hash Algorithm
SIFS	Short Interframe Space
SNR	Signal Noise Ratio
SoC	System On Chip
SPI	Synchronous Peripheral Interface
SPH	SPICLKOUT Phase
SPO	SPICLKOUT Polarity
SRAM	Static Random Access Memory
SSC	Spread Spectrum Clocking
SSCG	Spread Spectrum Clock Generator
SSI	Synchronous Serial Interface
SSS	Security Sub System
STA	Station
STBC	Space-Time Block Coding
SVB	Selective Voltage Binning
SVT	Standard Voltage Threshold
SWD	Serial Wire Debug

T

TI	Texas Instruments
TPC	Transmite Power Control
TRNG	True Random Number Generator
TSF	Timing Synchronization Function
TX	Transmitter

U

UAPSD	Unscheduled Automatic Power Save Delivery
UART	Universal Asynchronous Receiver & Transmitter
UPC	UP Converter
UVLO	Under Voltage Lock Out

V

VAP	Virtual Access Point
VCO	Voltage Controlled Oscillator
VGA	Variable Gain Amplifier
VHT	Very High Throughput
VSWR	Voltage Standing Wave Ratio

W

WADC	WiFi Analog Digital Converter
WAPI	WLAN Authentication and Privacy Infrastructure
WDAC	WiFi digital analog converter
WDT	Watch Dog Timer
WFA	Wi-Fi Alliance
WFI	Wait For Interrupt
WiFi	Wireless Fidelity
WIP	Write In Progress
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPS	Wi-Fi Protected Setup
WREN	Write Read Enable

WS	Word Select
X	
XIP	Executed In Place
XTAL	Quartz Crystal Unit