User Manual

APM32E103xCxE

Arm® Cortex® -M3 core-based 32-bit MCU

Version: V1.2

Contents

1 Introduction and Document Description Rules

1.1 Introduction

This *User Manual* provides application developers with all the information about how to use MCU (micro-controller) system architecture, memory and peripherals.

For information about Arm® Cortex®-M3 core, please refer to *Arm® Cortex®-M3 Technical Reference Manual*; please refer to the corresponding datasheet for detailed data such as model information, dimension and electrical characteristics of the device; for all MCU series models, please refer to the corresponding data manual for memory mapping, peripheral existence and their number.

1.2 Document Description Rules

1.2.1 "Register Functional Description" Rules

（1） Control (CTRL) registers are all "set 1 and clear 0 by software", unless otherwise specified.

（2） The control registers are usually followed by verb abbreviations to make a distinction. The verbs can be: EN-Enable, CFG-Configure, D-Disable, SET-Setup and SEL-Select

（3） The state register abbreviation is usually followed by FLG to make a difference.

（4） The value and data registers usually include V, VALUE, D and DATA, which are not followed by verbs, such as: xxPSC and CNT.

1.2.2 Full Name and Abbreviation Description of Terms

Table 1 Abbreviation and Description of R/W Modes

R/W mode	Description	Abbreviation
read/write	Software can read and write this bit.	R/W
read-only	Software can only read this bit.	R
write-only	Software can only write this bit, and after reading this bit, the reset value will be returned.	W
read/clear	The software can read this bit and clear it by writing 1. Writing 0 has no effect on this bit.	RC_W1
read/clear	The software can read this bit and clear it by writing 0. Writing 1 has no effect on this bit.	RC_W0
read/clear by read	The software can read this bit and reading this bit will automatically clear it to 0, and writing this bit is invalid.	RC_R

R/W mode	Description	Abbreviation
read/set	The software can read and set this bit, and writing 0 has no effect on this bit.	R/S
read-only write trigger	The software can read this bit and writing 0 or 1 can trigger an event but has no effect on the value of this bit.	RT_W
toggle	The software can flip this bit only by writing 1 and writing 0 has no effect on this bit.	T

Table 2 Functional Description and Full Name and Abbreviation of Terms of Commonly Used Registers

Full name in English	English abbreviation
Enable	EN
Disable	D
Clear	CLR
Select	SEL
Configure	CFG
Contrl	CTRL
Controller	C
Reset	RST
Stop	STOP
Set	SET
Load	LD
Calibration	CAL
Initialize	INIT
Error	ERR
Status	STS
Ready	RDY
Software	SW
Hardware	HW
Source	SRC
System	SYS
Peripheral	PER
Address	ADDR
Direction	DIR
Clock	CLK
Input	I

Full name in English	English abbreviation
Output	O
Interrupt	INT
Data	DATA
Size	SIZE
Divider	DIV
Prescaler	PSC
Multiplier	MUL
Period	PRD

Table 3 Full Name and Abbreviation of Modules

Full name in English	English abbreviation
External Memory Controller	EMMC
Static Memory Controller	SMC
Dynamic Memory Controller	DMC
Reset and Clock Management Unit	RCM
Power Management Unit	PMU
Backup Register	BAKPR
Nested Vector Interrupt Controller	NVIC
External Interrupt /Event Controller	EINT
Direct Memory Access	DMA
Debug MCU	DBG MCU
General-Purpose Input Output Pin	GPIO
Alternate Function Input Output Pin	AFIO
Timer	TMR
Watchdog Timer	WDT
Independent Watchdog Timer	IWDT
Windows Watchdog Timer	WWDT
Real-Time Clock	RTC
Universal Synchronous Asynchronous Receiver Transmitter	USART
Inter-Integrated Circuit Interface	I2C
Serial Peripheral Interface	SPI
Inter-IC Sound Interface	I2S
Quad Serial Peripheral Interface	QSPI

Full name in English	**English abbreviation**
Controller Area Network	CAN
Secure Digital Input and Output	SDIO
Universal Serial Bus Full-Speed Device	USBD
Analog-to-Digital Converter	ADC
Digital-to-Analog Converter	DAC
Cyclic Redundancy Check Calculation Unit	CRC
Float Point Unit	FPU

2 System Architecture

2.1 Full Name and Abbreviation Description of Terms

Table 4 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Advanced High-Performance Bus	AHB
Advanced Peripheral Bus	APB

2.2 System Architecture Block Diagram

The main system consists of four driving units and four passive units. The four driving units are DCode bus (D-bus), system bus (S-bus), general DMA1 and DMA2, which are connected to the core of Arm® Cortex®-M3. The four passive units are internal SRAM, internal flash memory, EMMC and bridge from AHB to APB (AHB2APBx). AHB2APBx connects all APB devices.

These are connected through a multi-level AHB bus architecture, as shown in the figure below:

Figure 1 APM32E103xCxE System Architecture Block Diagram

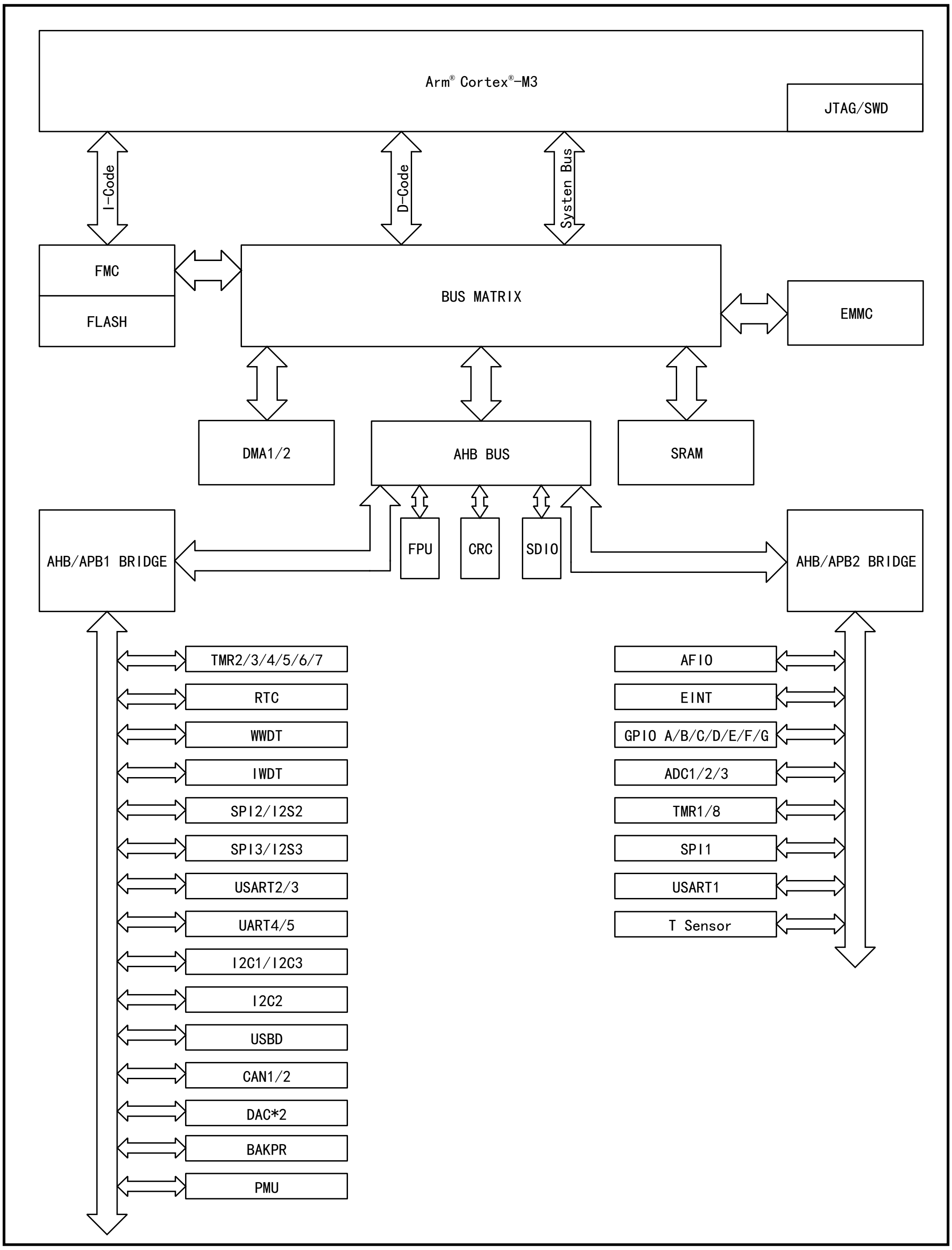
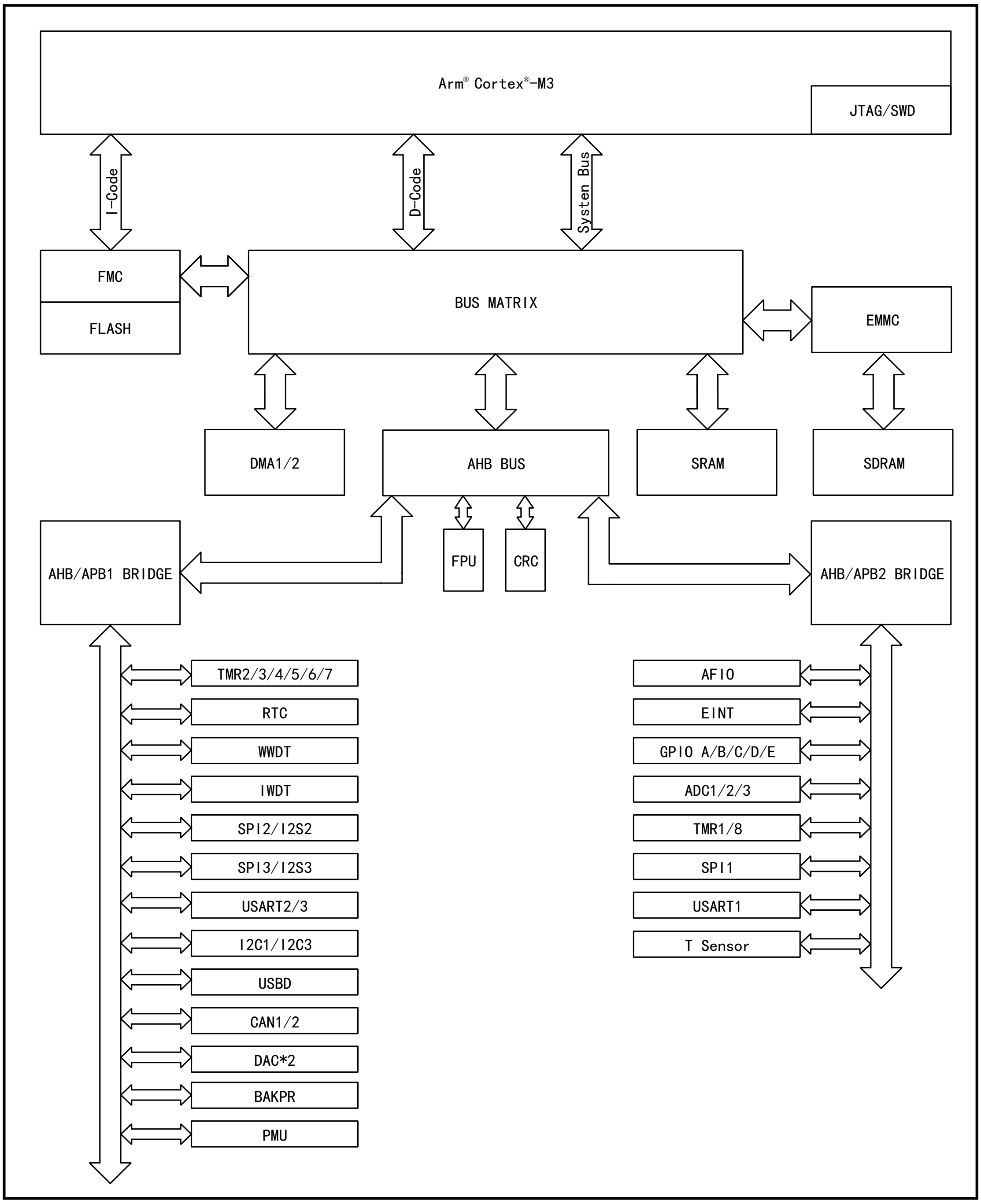


Figure 2 APM32E103VET6S System Architecture Block Diagram



Note:

(1) For details about the number of APM32E103xCxE modules of different models, see the datasheet.

(2) USBD1/2 use the same pin and address, so these two USBDs cannot be used at the same time, that is, only one USBD can be used at the same time.

Table 5 Bus Name

Name	Instruction
ICode Bus	Connect the instruction bus of Arm® Cortex®-M3 core and the flash instruction interface. Used for prefetched instructions.
DCode Bus	Connect the DCode bus of Arm® Cortex®-M3 core and the data interface of flash memory. Used for constants loading and access debugging.
System bus	Connect the system bus (peripheral bus) of Arm® Cortex®-M3 core and the bus matrix.
DMA bus	Connect AHB master control interface of DMA and the bus matrix.
Bus matrix	Coordinate the access to the core and DMA; coordinate the access of CPU's DCode and DMA to SRAM, Flash and peripherals. AHB peripheral is connected with the system bus through the bus matrix and is allowed to access DMA.
AHB/APB bridge	The two bridges provide a synchronous connection between AHB and the two APB buses. The maximum operating speed of APB1 and APB2 is different. The non-32-bit access to APB register will be converted into 32 bits automatically.

2.3 Memory Mapping

The memory mapping address is totally 4GB address. The assigned addresses include the core (including core peripherals), on-chip Flash (including main memory area, system memory area and option bytes), on-chip SRAM, EMMC and bus peripherals (including AHB and APB peripherals). Please refer to the data manual of the corresponding model for specific information of various addresses.

2.3.1 Embedded SRAM

Built-in static SRAM. It can access by byte, half word (16 bits) or full word (32 bits). The start address of SRAM is 0x2000 0000.

2.3.2 Bit band

Arm® Cortex®-M3 memory is mapped with two bit-band areas, and it maps each word in the alias memory area to one bit in the bit-band memory. Write a word to the alias memory and there will be the same effect as the read-change-write operation on the target of the bit-band area. Both peripheral register and SRAM are mapped into one bit band area, and it is allowed to perform single bit-band write and read operations.

The following gives a mapping formula:

bit_word_addr=bit_band_base+（byte_offset×32）+（bit_number×4）

Please see *Arm® Cortex®-M3 Technical Reference Manual* for details

2.4 Startup Configuration

Since the CPU of Arm® Cortex®-M3 core obtains reset vector from ICode Bus (instruction bus), the startup can only start from the code area, and the typical is Flash memory boot. However, APM32F MCU series realizes a special mechanism. By configuring the BOOT[1:0] pin parameters, there are three different startup modes, namely, the system can not only start from Flash memory or system memory, but also start from the built-in SRAM. The memory selected as the start zone is determined by the selected startup mode.

Table 6 Startup Mode Configuration and Access Mode

Startup mode selection pin		**Startup mode**	**Access mode**
BOOT1	**BOOT0**		
X	0	Main flash memory（Flash）	The main flash memory is mapped to the boot space, but it can still be accessed at its original address, that is, the contents of the flash memory can be accessed in two address areas.
0	1	System memory	The system memory is mapped to the boot space (0x0000 0000), but it can still be accessed at its original address.
1	1	Built-in SRAM	SRAM can be accessed only at the starting address.

Note:

(1) The boot space address is 0x0000 0000

(2) The original address of Flash is 0x0800 0000

(3) The original address of system memory is 0x1FFF F000

(4) The starting address of SRAM is 0x2000 0000

The user can select the startup mode after reset by setting the states of BOOT1 and BOOT0 pins. BOOT pin should keep the user's required startup configuration in standby mode. When exiting from the standby mode, the value of boot pin will be latched.

If you choose to start from built-in SRAM, you must use NVIC's exception table and offset register to remap the vector table to SRAM when writing the application code.

Embedded startup program

The embedded startup program is written on the production line by APEX and stored in the system memory area; the Flash memory can be reprogrammed by using USRAT1 to enable the startup program.

3 FLASH Memory

This chapter mainly introduces the storage structure, read, erase, write, read/write protection, unlock/lock characteristics of Flash, and the involved register functional description.

3.1 Full Name and Abbreviation Description of Terms

Table 7 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Flash Memory Controller	FMC

3.2 Main Characteristics

（1） Flash memory structure

- Contain main memory area and information block
- The capacity of main memory area is up to 512KB
- The information block is divided into system memory area and option byte
- The capacity of the system memory area is 2KB, for storing BootLoader program, 96-bit unique UID, and main memory area capacity information
- The capacity of the option byte area is 16Bytes

（2） Functional Description

- Read Flash
- Page/Card erase Flash
- Write Flash
- Read/Write protection Flash
- Configure option bytes

3.3 Flash Memory Structure

Table 8 Flash Memory Structure of APM32E103xCxE Series Products

Block	Name	Address range	Size (byte)
Main memory block	Page 0	0x0800 0000–0x0800 07FF	2K
Main memory block	Page 1	0x0800 0800–0x0800 0FFF	2K
Main memory block	Page 2	0x0800 1000–0x0800 17FF	2K
Main memory block	Page 3	0x0800 1800–0x0800 1FFF	2K
Main memory block	…	…	…
Main memory block	Page 255	0x0807 F800–0x0807 FFFF	2K

Block	Name	Address range	Size (byte)
Information block	System memory area	0x1FFF F000–0x1FFF F7FF	2K
Information block	Option byte	0x1FFF F800–0x1FFF F80F	16

Note: The number of pages in the main memory block of APM32E103xCxE series products is related to the Flash capacity of specific product.

3.4 Flash Memroy Functional Description

Describe the operation of main memory and information block (including system memory area and option byte), including read, write, erase and read/write protection.

Reading Flash includes main memory block and information block, while the erase, write, read/write are introduced separately; the system memory area has been written before the product leaves the factory and cannot be modified by the user. The erase, write, and read/write protection of the module will not be introduced.

3.4.1 Read Flash

Flash memory can be directly addressed, and reading Flash is affected by the following configuration:

Wait cycle

Different wait cycles should be configured for different system clocks:

- 0 wait cycle: 0<system clock≤24MHz
- 1 wait cycle: 24MHz<system clock≤48MHz
- 2 wait cycles: 48MHz<system clock≤72MHz
- 3 wait cycles: 72MHz<system clock≤96MHz
- 4 wait cycles: 96MHz<system clock≤120MHz

Prefetch buffer

It can improve the reading speed and every time it is reset, the prefetch buffer will be automatically opened. Only the system clock and AHB clock consistent and system clock less than 24 MHZ to configure open or closed.

Half-cycle access

When the power consumption needs to be optimized, half-cycle access can be used; at this time, the system clock and AHB clock are consistent, and the system clock is 8MHz or less than 8MHz, then half-cycle access to Flash can be used, otherwise, it must be turned on.

3.4.2 Main Memory Block

3.4.2.1 Erase main memory block

FMC supports page erase and mass erase (full erase) to initialize the contents of the main memory area to high level (the data is represented as 0xFFFF). Before writing to Flash, users are advised to erase the write address page. If the data of write address is not 0xFFFF, a programming error will be triggered.

Main memory page erase

Page erase is an independent erase according to the main memory area page selected by the program, which will not have any impact on the page not selected for erasure.

After the correct page erase (or flash write operation) is completed, OCF of FMC_STS register will be set. If OCIE interrupt is enabled, an operation completion interrupt will be triggered. Users need to note that the page to be erased must be a valid page (the valid address of the main memory area and the address not protected by write).

Main memory mass erase

The mass erase operation will erase all the contents in the main storage area of Flash, and the mass erase operation will erase all the data in the main memory area, so the users need to pay special attention when using it to avoid the loss of important data caused by misoperation.

3.4.2.2 Write Main memory block

FMC supports the writing of 16-bit (half word) data in the main memory area. You can select Debug, BootLoader, program running in SRAM, and directly reading the erased page to judge whether the erasing is successful.

In order to ensure correct writing, it is necessary to check whether the destination address has been erased before writing; if it is not erased, the written data will be invalid and PEF bit of FMC_STS register will be set to "1". If the destination address has write protection, the written data is invalid and a write protection error will be triggered (WPEF bit of FMC_STS is set to "1").

3.4.2.3 Main memory block of read/write protection

Read/Write protection of the flash is used to prevent illegal reading/modification of the main memory area code or data, and it is controlled by the read/write protection configuration byte of option byte. For APM32E103xCxE series products, the basic unit of read/write protection is 2 pages (i.e. 4KBytes).

Read protection

Internal Flash protection level can be set by modifying the value of option byte

READPROT. The debugger is always connected to JTAG/SWD interface to set read protection, which takes effect after power-on reset; otherwise, it will not take effect after the system is powered on and reset. When the READPROT value is any value except 0xA5, enable read protection and the content of main memory block cannot be read; when the READPROT value is 0xA5, the protection is released and the content of main memory block can be read; when the read protection is removed, a main memory mass erase operation will be triggered to prevent illegal read after the protection is degraded.

Write protection

Write protection control can be conducted for the corresponding page of the main memory block by configuring the value of write protection option byte WRP0/1/2/3. After the write protection is turned on, the content on the corresponding page of the main memory area cannot be modified in any way.

3.4.2.4 Main memory block of unlock/lock

FMC_CTRL1 of the reset FMC will be locked by hardware, and then FMC_CTRL1 can't be directly written, and the corresponding value must be written to FMC_KEY according to the correct sequence to unlock FMC. The KEY value is as follows:

- KEY1=0x45670123
- KEY2=0xCDEF89AB

The wrong writing sequence or wrong value will cause the program to enter the hardware wrongly. At this time, FMC will be locked, and all FMC operations will be invalid until it is reset next time. The users can also lock FMC through software by writing "1" to LOCK bit of the control register 2 (FMC_CTRL2).

In each Flash programming operation, the users must follow the steps of "Flash unlock - program by user - Flash lock", so as to avoid the risk that user code/data is accidentally modified due to the Flash unlocking after the Flash programming operation.

3.4.3 Option Byte

3.4.3.1 Erase option byte

Support erase function. After the correct option byte erase (or option byte write operation) is completed, OCF of FMC_STS register will be set. If OCIE interrupt is enabled, an operation completion interrupt will be triggered.

3.4.3.2 Write option byte

Eight configurable bytes of option bytes all support writing function.

3.4.3.3 Option byte of write protection

By default, the option byte is always readable and write protected. To perform write operation (program/erase) for the option byte block, first write the correct key sequence (the same as that of locking) in FMC_OBKEY, and then allow the write operation of option byte block; the OBWEN bit of FMC_CTRL2 register indicates write enabled; clear this bit and write operation will be disabled.

3.4.3.4 Unlock/Lock option byte

After the system reset, the option byte is locked by default. Only when the option byte is unlocked correctly, can it be modified. The difference between option byte unlocking and flash unlocking is that FMC_OBKEY register rather than FMC_KEY register writes the KEY value. The option byte does not support "software lock". The user should pay special attention to that every time after the value of the option byte is modified, the system must be reset to make it effective.

3.5 Option Byte Register Functional Description

The option byte provides some optional functions for users, and it mainly consists of 8 configurable bytes and corresponding complementary codes. Every time the system is reset, the option byte area will be reloaded to the FMC_ OBCS and FMC_WRTPROT register (the option byte will only take effect each time they are reloaded to FMC). In the process of reloading, if a certain configurable byte does not match its inverse code, an option byte error (OBE bit of FMC_OBCS register is set to "1") will be triggered, and this byte will be set to "0xFF". The information of 16 bytes in the option byte area is shown in the table below.

Table 9 Option Bytes

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F800	READPROT	0xA5	R/W	Read protection configuration
0x1FFF F801	nREADPROT	0x5A	R	READPROT complementary code
0x1FFF F802	UOB	0xFF	R/W	User option byte Bit 0: WDTSEL 0: Hardware watchdog 1: Software watchdog Bit 1: nRSTSTOP 0: Reset occurs when entering the Stop mode 1: Reset does not occur when entering the Stop mode Bit 2: nRSTSTB 0: Reset occurs when entering the Standby mode 1: Do not generate reset when entering Standby mode

Address	Option byte	Initial value	R/W	Functional description
				[3:7]: Reserved
0x1FFF F803	nUOB	0x00	R	UOB complementary code
0x1FFF F804	Data0	0xFF	R/W	User data byte 0
0x1FFF F805	nData0	0x00	R	Data0 complementary code
0x1FFF F806	Data1	0xFF	R/W	User data byte 1
0x1FFF F807	nData1	0x00	R	Data complementary code
0x1FFF F808	WRP0	0xFF	R/W	Write protection configuration 0
0x1FFF F809	nWRP0	0x00	R	WRP0 complementary code
0x1FFF F80A	WRP1	0xFF	R/W	Write protection configuration 1
0x1FFF F80B	nWRP1	0x00	R	WRP1 complementary code
0x1FFF F80C	WRP2	0xFF	R/W	Write protection configuration 2
0x1FFF F80D	nWRP2	0x00	R	WRP2 complementary code
0x1FFF F80E	WRP3	0xFF	R/W	Write protection configuration 3
0x1FFF F80F	nWRP3	0x00	R	WRP3 complementary code

Note: When the configurable byte and its complementary value are "0xFF", the match will not be verified in the reloading process

Table 10 Write Protection WRPx Function Description of Main Memory Area

Product capacity	Functional description
APM32E103xCxE series products (2KB/page)	Each bit in WRPx controls the write protection of 4KB (2 pages) address of the main memory area 0: Write protection is turned on 1: Write protection is not turned on WRP0: Page 0-15 WRP1: Page 16-31 WRP2: Page 32-47 WRP3: Bit0-Bit6 controls Page 48-61; Bit7 controls Page 62-255

Note: Flash read/write protection configuration is independent of each other. Removing the write protection will not force the loss of the contents of the main memory area, but keep them as they are.

3.6 FMC Register Address Mapping

Base address: 0x40022000

Table 11 FMC Register Address Mapping

Register name	Description	Offset address
FMC_CTRL1	Control register 1	0x00
FMC_KEY	Key register	0x04
FMC_OBKEY	Option byte register	0x08

Register name	Description	Offset address
FMC_STS	State register	0x0C
FMC_CTRL2	Control register 2	0x10
FMC_ADDR	Flash address register	0x14
FMC_OBCS	Option byte control/state register	0x1C
FMC_WRTPROT	Write protection register	0x20

3.7 FMC Register Functional Description

3.7.1 Control register 1 (FMC_CTRL1)

Offset address: 0x00

Reset value: 0x0000 0030

Field	Name	R/W	Description
2:0	WS	R/W	Wait State Configure 000: 0 wait cycle, 0<system clock≤24MHz 001: 1 wait cycle: 24MHz<system clock≤48MHz 010: 2 wait cycles, 48MHz<system clock≤72MHz 011: 3 wait cycles, 72MHz<system clock≤96MHz 100: 4 wait cycles, 96MHz<system clock≤120MHz Others: Reserved
3	HCAEN	R/W	Flash Half Cycle Access Enable 0: Disable 1: Enable
4	PBEN	R/W	Prefetch Buffer Enable 0: Disable 1: Enable
5	PBSF	R	Prefetch Buffer Status Flag 0: In disabled state 1: In enabled state
31:6	Reserved		

3.7.2 Key register (FMC_KEY)

Offset address: 0x04

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:0	KEY	W	FMC Key Writing the keys represented by these bits can unlock FMC. These bits can only perform write operation, and 0 is returned when read operation is performed.

3.7.3 Option byte key register (FMC_OBKEY)

Offset address: 0x08

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:0	OBKEY	W	Option Byte Key Writing the keys represented by these bits can unlock the option byte write operation. These bits can only perform write operation and 0 is returned when read operation is performed.

3.7.4 State register (FMC_STS)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	BUSYF	R	Busy Flag This bit indicates that a flash operation is in progress. These bits can only perform write operation, and 0 is returned when read operation is performed.
1	Reserved		
2	PEF	R/W	Programming Error Flag This bit will be set by software when the value before the address is edited is not "0xFFFF".
3	Reserved		
4	WPEF	R/W	Write protection error flag (Write Protection Error Flag) This bit will be set by hardware when programming the write protection address in FLASH.
5	OCF	R/W	Operation completion flag (Operation Complete Flag) This bit will be set by hardware when read/write operation in FLASH is completed.
31:6	Reserved		

3.7.5 Control register 2 (FMC_CTRL2)

Offset address: 0x10

Reset value: 0x0000 0080

Field	Name	R/W	Description
0	PG	R/W	Program Set this bit to 1 to program Flash
1	PAGEERA	R/W	Page Erase Set this bit to 1 to erase the page
2	MASSERA	R/W	Mass Erase Set this bit to 1 to erase the mass.
3	Reserved		
4	OBP	R/W	Option Byte Program Set this bit to 1 to program the option byte.
5	OBE	R/W	Option Byte Erase Set this bit to 1 to erase the option byte.
6	STA	R/W	Start Erase This bit can be only set to 1 by software, and can be reset by clearing STS_BUSYF bit.

Field	Name	R/W	Description
7	LOCK	R/W	Lock This bit can be written to 1 only, and when this bit is set to 1, it means that FMC and CTRL2 registers are locked.
8	Reserved		
9	OBWEN	R/W	Option Byte Write Enable When this bit is set to 1, the option byte can be programmed.
10	ERRIE	R/W	Error interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled When STS_PEF=1 or STS_WPEF=1, set this bit to generate an interrupt.
11	Reserved		
12	OCIE	R/W	Operation Complete Interrupt Enable 0: Operation completion interrupt is disabled 1: Operation completion interrupt is enabled When STS_OCF=1, set this bit to generate an interrupt.
31:13	Reserved		

3.7.6 Flash address register (FMC_ADDR)

Offset address: 0x14

Reset value: 0x0000 0000

The register is changed to currently/finally used address by hardware; in page erasing, the register needs to be configured by software.

Field	Name	R/W	Description
31:0	ADDR	W	Flash Address In programming operation, the bit is written to the address to be programmed; in page erasing, this bit is written to the page to be erased.

3.7.7 Option bye control/state register (FMC_OBCS)

Offset address: 0x1C

Reset value: 0x03FF FFFC

The reset value of the register is related to the value in the written option byte; the reset value of OBE bit is related to the result whether the value of the loaded option byte is consistent with its reverse code.

Field	Name	R/W	Description
0	OBE	R	Option Byte Error 1: The loaded option byte does not match its complementary code. The option byte and its complementary code are forced to write to 0xFF
1	READPROT	R	Read Protect 1: Indicate that the flash memory is in read protection state

Field	Name	R/W	Description
9:2	UOB	R	User Option Byte Here include the user option byte loaded by OBL Bit 2: WDTSEL Bit 3: RSTSTOP Bit 4: RSTSTDB Bit [9:5]: Unused
17:10	DATA0	R	Data0
25:18	DATA1	R	Data1
31:26	Reserved		

3.7.8 Write protection register (FMC_WRTPROT)

Offset address: 0x20

Reset value: 0xFFFF FFFF

Field	Name	R/W	Description
31:0	WRTPROT	R	Write Protect 0: Valid 1: Invalid

4 External Memory Controller (EMMC)

4.1 Full Name and Abbreviation Description of Terms

Table 12 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Static Random Access Memory	SRAM
Read Only Memory	ROM
Pseudo Static Random Access Memory	PSRAM
Random Access Memory	RAM
Synchronous Dynamic Random Access Memory	SDRAM
Multiplex	MUX
Width	WID
Flash Memory	FM
Access	ACC
Wait	W
Signal	S
Polarity	POL
Asynchronous	ASYN
Burst	BURST
Timing	TIM
Setup	SET
Hold	HLD
Empty	E

4.2 EMMC Overview

EMMC includes SMC (static memory controller) and DMC (dynamic memory controller). SMC is responsible for controlling SRAM, PSRAM, NandFlash, NorFlash and PCCard; DMC is responsible for controlling SDRAM.

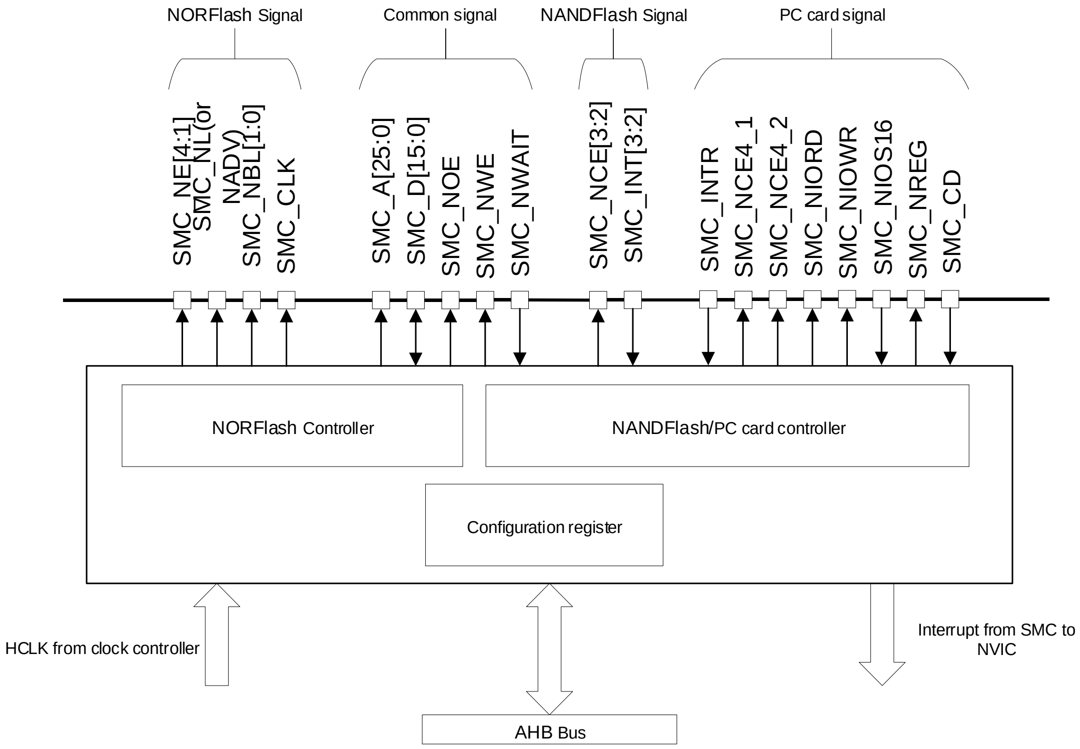
4.3 SMC Introduction

SMC is used to manage the extended static memory peripherals; AHB transmission signals can be converted to the appropriate external devices; there are four internal memory blocks, each of which controls different types of memory and is distinguished by chip selection signal; only one external device can be accessed at any moment; each memory block can be configured separately, and the timing can be programmed for external devices.

4.4 SMC Structure Block Diagram

SMC consists of five parts: AHB bus interface, configuration register, NORFlash controller, NANDFlash/PC card controller and external device interface, specifically as shown in the figure below:

Figure 3 SMC Block Diagram



4.5 SMC Functional Description

4.5.1 SMC Access Rules

SMC is an interface for internal CPU to access external static memory through AHB bus. On AHB bus, the operation of 32-bit data will be converted into continuous 16 or 8-bit operation. In order to ensure the consistency of data transmission, SMC needs to comply with the following rules in external read-write operation:

（1） When the width of AHB accessing external data is equal to that of memory data, the data can be transmitted normally without any problem.

（2） When the width of AHB accessing external data is larger than that of memory data, the access operation will be automatically cut to be consistent with the width of external data for transmission.

（3） When the width of AHB accessing external data is less than that of memory data, if the external memory has the function of byte selection, it can transmit data normally through byte channel; if it does not have such function, it does not allow write operation, and only allows read operation.

4.5.2 External Device Address Mapping

SMC divides external devices into multiple memory blocks, and different memory blocks control different external devices. The specific classification is shown in the table below:

Table 13 External Device Address Mapping Table

Start address	End address	Memory block	Memory type supported
0x60000000	0x6FFFFFFF	Memory block 1 (4*64MB)	NOR/PSRAM
0x70000000	0x7FFFFFFF	Memory block 2 (4*64MB)	NAND
0x80000000	0x8FFFFFFF	Memory block 3 (4*64MB)	NAND
0x90000000	0x9FFFFFFF	Memory block 4 (4*64MB)	PC card

4.5.3 NOR Flash Memory and PSRAM

4.5.3.1 Address mapping

Memory block 1 is used to control NOR/PSRAM memory, which is divided into four 64MB areas of the same size. The selection of each area depends on the value of HADDR[27:26], and the specific information is as follows:

Table 14 Address Mapping of Memory Block 1

HADDR[27:26]	Start address	End address	Area block
00	0x60000000	0x63FFFFFF	Area block 1
01	0x64000000	0x67FFFFFF	Area block 2
10	0x68000000	0x6BFFFFFF	Area block 3
11	0x6C000000	0x6FFFFFFF	Area block 4

HADDR is the internal AHB address line that needs to be converted to the external memory. It is a byte address. However, some external memories are not accessed by byte, so the address may be inconsistent. In order to avoid the error caused by this situation, SMC will be adjusted according to the following rules:

- When the width of external memory data is 8 bits, HADDR[25:0] is connected to SMC_ A [25:0], while SMC_A[25:0] is connected to the external memory address line.

- When the width of external memory data is 16 bits, HADDR[25:1] is connected to SMC_ A [24:0], while SMC_A[24:0] is connected to the external memory address line.

4.5.3.2 Interface signal and controler

The memory block 1 supports NOR Flash, PSRAM, SRAM and ROM external memory. There are corresponding chip selection signals NE[x] (x=1..4) in the four areas of the memory block 1. All other signals are shared. The specific interface signals and functions are as follows:

Table 15 NOR Flash Interface Signal

SMC signal name	Signal direction	Function
CLK	Output	Synchronous clock signal
NE[x]	Output	Chip selection signal, x=1..4
NOE	Output	Read enable signal
NEW	Output	Write enable signal
NWAIT	Input	Signal that NOR flash memory requires SMC to wait
A[25:0]	Output	Non-multiplexing: A[25:0] all are address bus
	Output	Multiplexing: A[25:16] is address bus
AD[15:0]	Input/Output	Non-multiplexing: Bidirectional data bus
	Input/Output	Multiplexing: Bidirectional address/Data bus
NL (=NADV)	Output	Effective address signal

Table 16 PSRAM Interface Signal

SMC signal name	Signal direction	Function
CLK	Output	Synchronous clock signal
NE[x]	Output	Chip selection signal, x=1..4
NOE	Output	Read enable signal
NEW	Output	Write enable signal
NWAIT	Input	Signal that PSRAM requires SMC to wait
A[25:0]	Output	Address bus
D[15:0]	Input/Output	Bidirectional data bus
NL（=NADV）	Output	Effective address signal
NBL[1]	Output	High byte enable
NBL[0]	Output	Low byte enable

Note: The output signal of the controller changes at the rising edge of the internal clock; in the synchronous write mode, the output data changes at the falling edge of the memory clock.

NOR Flash/PSRAM controller provides programmable timing parameters for external memory, including the parameters in the following table:

Table 17 Programmable NOR/PSRAM Timing Parameters

Parameter	Function	Access mode	Unit	Minimum	Maximum
Data generation time	The number of clocks required to generate the first data in burst mode	Synchronous	Memory clock cycle (CLK)	2	17
Clock division factor	The ratio of memory access clock cycle (CLK) to AHB clock cycle	Synchronous	AHB clock cycle (HCLK)	2	16
Bus recovery time	Duration of Bus recovery phase	Asynchronous or synchronous read		1	16
Data setup time	Duration of data setup phase	Asynchronous		2	256
Address hold time	Duration of address hold phase	Synchronous, multiplexing IO		2	16
Address setup time	Duration of address setup phase	Asynchronous		1	16

4.5.4 NAND Flash Memory and PC Card

4.5.4.1 Address mapping

Memory blocks 2, 3 and 4 are used to access NAND flash memory and PC card. Each memory block is also divided into different areas, the corresponding effect of different areas is different, and the specific distribution is as follows:

Table 18 Address Mapping of Memory Blocks 2、3 and 4

SMC memory block	Storage space	Start address	End address
Memory block 2-NAND flash memory	General	0x70000000	0x73FFFFFF
	Attributes	0x78000000	0x7BFFFFFF
Memory block 3-NAND flash memory	General	0x80000000	0x83FFFFFF
	Attributes	0x88000000	0x8BFFFFFF
Memory block 4-PC card	General	0x90000000	0x93FFFFFF
	Attributes	0x98000000	0x9BFFFFFF

SMC memory block	Storage space	Start address	End address
	I/O	0x9C000000	0x9FFFFFFF

NAND flash memory block is divided into three blocks in part of the low-byte area, and different blocks can be accessed through HADDR[17:16]. The specific division and selection of these three blocks are shown in the table below:

Table 19 NAND Memory Block Division

HADDR[17:16]	Address range	Block name
00	0x000000-0x00FFFF	Data block
01	0x010000-0x01FFFF	Command block
1X	0x020000-0x03FFFF	Address block

In order to read and write NAND memory normally, the following steps are needed:

- Send command to the memory
- Send the address for reading and writing to the memory
- Read/Write data

The operation address of the corresponding three-step operation corresponds to the three blocks in the memory block. To send a command to the memory is to write the corresponding command value to the command block; to send an address to the memory is to send the corresponding address value to the address block; to read and write data is to read and write in the data block; finally write or read out the internal unit of NAND, and the address of the corresponding unit is the address written in the address block.

4.5.4.2 Interface signal and controler

NAND/PC card controller can control three memory blocks. The memory blocks 2 and 3 support NAND Flash, and the memory block 4 supports PC card devices. Three memory blocks have their own chip selection signals, and the specific interfaces and functions are as follows:

Table 20 NAND Flash Interface Signal

SMC signal name	Signal direction	Function
NCE[x]	Output	Chip selection signal, x=2, 3
NOE（=NRE）	Output	Read enable signal
NEW	Output	Write enable signal
NWAIT/INT[3:2]	Input	NAND Flash ready/busy input signal
A[17]	Output	NAND Flash address latch signal (ALE)
A[16]	Output	NAND Flash command latch signal (CLE)

SMC signal name	Signal direction	Function
D[15:0]	Input/Output	8-bit multiplexing: D[7:0] bidirectional address/data bus
	Input/Output	16-bit multiplexing: D[15:0] bidirectional address/data bus

Table 21 PC Card Interface Signal

SMC signal name	Signal direction	Function
NCE4_1	Output	Chip selection signal 1
NCE4_2	Output	Chip selection signal 2 (select 16-bit or 8-bit operation)
NOE	Output	Read enable signal
NEW	Output	Write enable signal
NWAIT	Input	PC card wait signal
INTR	Input	PC card interrupt signal
CD	Input	PC card detection signal
A[10:0]	Output	Address bus
NIOS16	Input	Data transmission width of 16-bit transmission I/O space (must be grounded)
NIORD	Output	I/O space output enable
NIOWR	Output	I/O space write enable
NREG	Output	Selection of common space or attribute space access
D[15:0]	Output/Input	Bidirectional data bus

NAND Flash/PC card controller provides programmable timing parameters for external memory, including the parameters in the following table:

Table 22 Programmable NAND/PC Timing Parameters

Parameter	Function	Operation mode	Unit	Minimum	Maximum
Memory data bus high-impedance time	The time of holding the data bus in high-impedance state after starting write operation	Write	AHB clock cycle (HCLK)	0	255
Memory hold time	The number of clocks holding the address after sending the command, also the hold time of data during write operation	Read/Write		1	254

Parameter	Function	Operation mode	Unit	Minimum	Maximum
Memory waiting time	Minimum sending duration			2	256
Memory setup time	The number of clocks that set up the address before sending the command			1	255

4.6 SMC Register Address Mapping

Table 23 SMC Register Address Mapping

Register name	Description	Offset address
SMC_CSCTRL1…4	SRAM/NOR flash memory chip selection control register 1…4	8*(x-1),x=1…4
SMC_CSTIM1…4	SRAM/NOR flash memory chip selection timing register 1…4	0x04 + 8*(x-1),x=1…4
SMC_WRTTIM1…4	SRAM/NOR flash memory write timing register 1…4	0x104 + 8*(x-1),x=1…4
SMC_CTRL2…4	PC card/NAND flash memory control register 2…4	0x40 + 0x20 * (x-1),x=2…4
SMC_STSINT2…4	FIFO state and interrupt register 2…4	0x44 + 0x20 * (x-1),x=2…4
SMC_CMSTIM2…4	General-purpose memory space timing register 2...4	0x48 + 0x20 * (x-1),x=2…4
SMC_AMSTIM2…4	Attribute memory space timing register 2...4	0x4C + 0x20 * (x-1),x=2…4
SMC_IOSTIM4	I/O space timing register 4	0XB0
SMC_ECCRS2/3	ECC result register 2/3	0x54 + 0x20 * (x-1), x=2 or 3

4.7 SMC Register Functional Description

4.7.1 NOR Flash Memory and PSRAM Control Register

4.7.1.1 SRAM/NOR flash memory chip selection control register 1…4 (SMC_CSCTRL1…4)

Offset address: 8*(x-1), x=1…4

Reset value: 0x0000 30DX

Field	Name	R/W	Description
0	MBKEN	R/W	Enable the Corresponding Memory Bank 0: Disable 1: Enable

Field	Name	R/W	Description
1	ADMUXEN	R/W	Address/Data Multiplexing Enable This bit is effective only for NORFlash and PSRAM. 0: Disable 1: Address low 16-bit and data sharing data bus
3:2	MTYPECFG	R/W	Memory Type Configure 00: SRAM, ROM (default value of Bank2~Bank4 after reset) 01：PSRAM 10: NORFlash (default value after Bank1 reset) Others reserved
5:4	MDBWIDCFG	R/W	Memory Data Bus Width Configure 00: 8 bits 01: 16 bits Others reserved
6	NORFMACCEN	R/W	NORFlash Memory Access Enable 0: Disable 1: Enable
7	Reserved		
8	BURSTEN	R/W	Burst Mode Enable In synchronous mode, use the burst mode to access the memory. 0: Disable 1: Enable
9	WSPOLCFG	R/W	Wait Signal Polarity Configure This bit is effective only in burst mode. 0: Low effective 1: High effective
10	WRAPBEN	R/W	Wrapped Burst Mode Enable This bit is effective only in burst mode. 0: Not allowed 1: Allowed
11	WTIMCFG	R/W	Wait Timing Configure This bit is used to configure whether the memory generates NWAIT signal in the period before the waiting state or during the wait cycle; this bit is effective only in burst mode. 0: NWAIT signal is effective in the data cycle before waiting 1: NWAIT signal is effective in the wait cycle
12	WREN	R/W	Write Memory Enable This bit is used to enable write operation of SMC for the memory. 0: Disable write; otherwise, an AHB error will be generated 1: Allow write
13	WAITEN	R/W	Wait Enable This bit is used to enable NWAIT signal to insert the wait state; this bit is effective only in burst mode. 0: Disable 1: Enable

Field	Name	R/W	Description
14	EXTMODEEN	R/W	Extended Mode Enable In extended mode, SMC_WRTTIM register can be used to realize read and write using different timing function. 0: Disable 1: Use
15	WSASYNCEN	R/W	Wait Signal During Asynchronous Transfers Enable This bit is used to enable SMC to use NWAIT signal during asynchronous protocol period. 0: Disable 1: Enable
18:16	CRAMPSIZECFG	R/W	CRAM Page Size Configure 000: There is no burst split when crossing the page boundary 001: 128 bytes 010: 256 bytes 011: 512 bytes 100: 1024 bytes Others: Reserved
19	WRBURSTEN	R/W	Write PSRAM Burst Enable This bit is used to enable the synchronous burst transmission protocol for write operation. 0: Write operation is asynchronous mode 1: Write operation is synchronous mode
31:20	Reserved		

4.7.1.2 SRAM/NOR flash memory chip selection timing register 1…4 (SMC_CSTIM1…4)

Offset address: 0x04 + 8*(x-1),x=1…4

Reset value: 0x0FFF FFFF

Field	Name	R/W	Description
3:0	ADDRSETCFG	R/W	Address Setup Time Configure Only apply to NOR flash memory operation in SRAM, ROM and asynchronous bus multiplexing mode. 0000: 1 HCLK clock cycle 0001: 2 HCLK clock cycles …… 1111: 16 HCLK clock cycles Note: In synchronous operation, this parameter is meaningless and is always one memory clock cycle

Field	Name	R/W	Description
7:4	ADDRHLDCFG	R/W	Address-Hold Time Configure Only apply to NOR flash memory operation in SRAM, ROM and asynchronous bus multiplexing mode. 0000: Reserved 0001: 2 HCLK clock cycles …… 1111: 16 HCLK clock cycles Note: In synchronous operation, this parameter is meaningless and is always one memory clock cycle
15:8	DATASETCFG	R/W	Data Setup Time Configure Only apply to NOR flash memory operation in SRAM, ROM and asynchronous bus multiplexing mode. 0000 0000: Reserved 0000 0001: 2 HCLK clock cycles 0000 0010: 3 HCLK clock cycles …… 1111 1111: 256 HCLK clock cycles
19:16	BUSTURNCFG	R/W	Bus Turnaround Phase Duration Configure These bits are used to configure the delay time on the bus after a read operation. They are only applicable to NOR flash memory operation in bus multiplexing mode. 0000: 1 HCLK clock cycle 0001: 2 HCLK clock cycles …… 1111: 16 HCLK clock cycles
23:20	CLKDIVCFG	R/W	Clock Divide Factor Configure CLK comes from HCLK frequency division. These bits are used to configure the frequency of CLK clock output signal. They are only applicable to synchronous mode. 0000: Reserved 0001: 2 divided frequency 0010: 3 divided frequency …… 1111: 16 divided frequency Note: This parameter is ineffective when accessing asynchronous NOR flash memory, SRAM or ROM.
27:24	DATALATCFG	R/W	Data Latency Configure These bits are used to configure the number of memory cycles for waiting before reading the first data. They are only applicable to NOR flash memory operation in synchronous burst mode. 0000: 2 CLK clock cycles 0001: 3 CLK clock cycles …… 1111: 17 CLK clock cycles Note: When accessing asynchronous NOR flash memory, SRAM or ROM, this parameter is invalid. When operating CRAM, this parameter is 0.

Field	Name	R/W	Description
29:28	ASYNCACCCFG	R/W	Asynchronous Access Mode Configure Valid only when EXTMODEEN bit of SMC_CSCTRLX register is 1. 00: Access mode A 01: Access mode B 10: Access mode C 11: Access mode D
31:30	Reserved		

4.7.1.3 SRAM/NOR flash memory write timing register 1…4 (SMC_WRTTIM1…4)

Offset address: 0x104 + 8*(x-1), x=1…4

Reset value: 0x0FFF FFFF

Field	Name	R/W	Description
3:0	ADDRSETCFG	R/W	Address Setup Time Configure Only apply to NOR flash memory operation in SRAM, ROM and asynchronous bus multiplexing mode. 0000: 1 HCLK clock cycle 0001: 2 HCLK clock cycles …… 1111: 16 HCLK clock cycles Note: In synchronous operation, this parameter is meaningless and is always 1 memory clock cycle
7:4	ADDRHLDCFG	R/W	Address-Hold Time Configure Only apply to NOR flash memory operation in SRAM, ROM and asynchronous bus multiplexing mode. 0000: Reserved 0001: 2 HCLK clock cycles …… 1111: 16 HCLK clock cycles Note: In synchronous operation, this parameter is meaningless and is always 1 memory clock cycle
15:8	DATASETCFG	R/W	Data-Setup Time Configure Only apply to NOR flash memory operation in SRAM, ROM and asynchronous bus multiplexing mode. 0000 0000: Reserved 0000 0001: 2 HCLK clock cycles 0000 0010: 3 HCLK clock cycles …… 1111 1111: 256 HCLK clock cycles
19:16	BUSTURNCFG	R/W	Bus Turnaround Phase Duration Configure These bits are used to configure the delay time on the bus after a read operation. They are only applicable to NOR flash memory operation in bus multiplexing mode. 0000: 1 HCLK clock cycle 0001: 2 HCLK clock cycles …… 1111: 16 HCLK clock cycles

Field	Name	R/W	Description
23:20	CLKDIVCFG	R/W	Clock divide ratio (for CLK signal Configure) Cycle of CLK clock output signal, expressed by the number of HCLK cycles. 0000: Reserved 0001: CLK cycle=2×HCLK cycle 0010: CLK cycle =3×HCLK cycle 1111: CLK cycle=16×HCLK cycle (default value after reset) This bit is invalid in asynchronous NOR Flash, SRAM or ROM access mode.
27:24	DATLATCFG	R/W	Data latency Configure This bit is used to define the memory clock to be transmitted to the memory before obtaining the first data Period number (+2). 0000: Data delay of 2 CLK clock cycles during the first burst access … 1111: Data delay of 17 CLK clock cycles during the first burst access (default value after reset) This bit is invalid in asynchronous NOR Flash, SRAM or ROM access mode.
29:28	ASYNCACCCFG	R/W	Asynchronous Access Mode Configure These bits are used to configure asynchronous access mode, and are valid only when EXTMODEEN bit of SMC_CSCTRLX register is 1. 00: Access mode A 01: Access mode B 10: Access mode C 11: Access mode D
31:30	Reserved		

4.7.2 NAND Flash Memory and PC Card Control Register

4.7.2.1 PC card/NAND flash memory control register 2…4 (SMC_CTRL2…4)

Offset address: 0x40 + 0x20 * (x-1), x=2…4

Reset value: 0x0000 0018

Field	Name	R/W	Description
0	Reserved		
1	WAITFEN	R/W	PC Card/NANDFlash Wait Feature Enable 0: Disable 1: Enable
2	MBKEN	R/W	PC Card/NAND Flash Memory Bank Enable 0: Disable 1: Enable
3	MTYPECFG	R/W	Memory Type Configure 0: PC card, CF card, CF+ card or PCMCIA 1: NAND flash memory

Field	Name	R/W	Description
5:4	DBWIDCFG	R/W	Databus Width Configure 16 bits must be used for PC Card. 00: 8 bits 01: 16 bits Others reserved
6	ECCEN	R/W	ECC Computation Logic Enable 0: Disable and reset ECC 1: Enable
8:7	Reserved		
12:9	C2RDCFG	R/W	CLE To RE Delay Configure Configure the duration from "CLE becomes low level" to "RE becomes low level". 0000: 1 HCLK cycle 0000: 2 HCLK cycles …… 1111: 16 HCLK cycles
16:13	A2RDCFG	R/W	ALE To RE Delay Configure Configure the duration from "ALE becomes low level" to "RE becomes low level" 0000: 1 HCLK cycle 0000: 2 HCLK cycles …… 1111: 16 HCLK cycles
19:17	ECCPSCFG	R/W	ECC Page Size Configure 000: 256 bytes 001: 512 bytes 010: 1024 bytes 011: 2048 bytes 100: 4096 bytes 101: 8192 bytes
31:20	Reserved		

4.7.2.2 FIFO state and interrupt register 2…4 (SMC_STSINT2…4)

Offset address: 0x44 + 0x20 * (x-1), x=2…4

Reset value: 0x0000 0040

Field	Name	R/W	Description
0	IREFLG	R/W	Interrupt Rising Edge Generate Flag This bit is set to 1 by hardware and cleared by software. 0: Not generate 1: Generate
1	IHLFLG	R/W	Interrupt High-Level Generate Flag This bit is set to 1 by hardware and cleared by software. 0: Not generate 1: Generate

Field	Name	R/W	Description
2	IFEFLG	R/W	Interrupt Falling Edge Generate Flag This bit is set to 1 by hardware and cleared by software. 0: Not generate 1: Generate
3	IREDEN	R/W	Interrupt Rising Edge Detection Enable 0: Disable 1: Enable
4	IHLDEN	R/W	Interrupt High-Level Detection Enable 0: Disable 1: Enable
5	IFEDEN	R/W	Interrupt Falling Edge Detection Enable 0: Disable 1: Enable
6	FEFLG	R	FIFO Empty Flag 0: Not empty 1: Empty
31:7	Reserved		

4.7.2.3 General-purpose memory space timing register 2…4 (SMC_CMSTIM2…4)

Offset address: 0x48 + 0x20 * (x-1), x=2…4

Reset value: 0xFCFC FCFC

Field	Name	R/W	Description
7:0	SETx	R/W	Common Memory x Setup Time Configure This bit takes CLK as the clock cycle, and defines the time of setting up the address before sending the command. 0000 0000: 1 HCLK cycle 0000 0001: 2 HCLK cycles …… 1111 1110: 255 HCLK cycles 1111 1111: Reserved
15:8	WAITx	R/W	Common Memory x Wait Time Configure This bit takes HCLK as the clock cycle and defines the minimum hold time of the command. After the defined time, if the waiting signal is effective low, the hold time of the command will become longer. 0000 0000: Reserved 0000 0001: 2 HCLK cycles(+ wait cycle introduced by deasserting NWAIT) 0000 0010: 3 HCLK cycles …… 1111 1110: 255 HCLK cycles(+ wait cycle introduced by deasserting NWAIT) 1111 1111: Reserved

Field	Name	R/W	Description
23:16	HLDx	R/W	Common Memory x Hold Time Configure This bit takes CLK as the clock cycle, and defines the hold time of address signal after sending the command. 0000 0000: Reserved 0000 0001: 1 HCLK cycle for write accesses, 3 HCLK cycles for read accesses 0000 0010: 2 HCLK cycles …… 1111 1110: 254 HCLK cycles for write accesses, 256 HCLK cycles for read accesses 1111 1111: Reserved
31:24	HIZx	R/W	Common Memory x Databus Hiz Time Configure This bit takes HCLK as the clock cycle and defines the time of high-impedance state of data bus, which is only effective for write operation. 0000 0000: 1 HCLK cycle 0000 0001: 2 HCLK cycles …… 1111 1110: 255 HCLK cycles 1111 1111: Reserved

4.7.2.4 Attribute memory space timing register 2…4 (SMC_AMSTIM2…4)

Offset address: 0x4C + 0x20 * (x-1), x=2…4

Reset value: 0xFCFC FCFC

Field	Name	R/W	Description
7:0	SETx	R/W	Attribute Memory x Setup Time Configure This bit takes CLK as the clock cycle, and defines the time of setting up the address signal before sending the command. 0000 0000: 1 HCLK cycle 0000 0001: 2 HCLK cycles …… 1111 1110: 255 HCLK cycles 1111 1111: Reserved
15:8	WAITx	R/W	Attribute Memory x Wait Time Configure This bit takes HCLK as the clock cycle and defines the minimum hold time of the command. After the defined time, if the waiting signal is effective low, the hold time of the command will become longer. 0000 0000: Reserved 0000 0001: 2 HCLK cycles(+ wait cycle introduced by deasserting NWAIT) 0000 0010: 3 HCLK cycles …… 1111 1110: 255 HCLK cycles(+ wait cycle introduced by deasserting NWAIT) 1111 1111: Reserved

Field	Name	R/W	Description
23:16	HLDx	R/W	Attribute Memory x Hold Time Configure This bit takes CLK as the clock cycle, and defines the hold time of address signal after sending the command. 0000 0000: Reserved 0000 0001: 1 HCLK cycle for write accesses, 3 HCLK cycles for read accesses 0000 0010: 2 HCLK cycles …… 1111 1110: 254 HCLK cycles for write accesses, 256 HCLK cycles for read accesses 1111 1111: Reserved
31:24	HIZx	R/W	Attribute Memory x Databus Hiz Time Configure This bit takes HCLK as the clock cycle and defines the time of high-impedance state of data bus, which is only effective for write operation. 0000 0000: 0 HCLK cycle 0000 0001: 1 HCLK cycle …… 1111 1111: 255 HCLK cycles

4.7.2.5 I/O space timing register 4 (SMC_IOSTIM4)

Offset address: 0xB0

Reset value: 0xFCFC FCFC

Field	Name	R/W	Description
7:0	SET	R/W	I/O x Setup Time Configure This bit takes CLK as the clock cycle, and defines the time of setting up the address signal before sending the command. 0000 0000: 1 HCLK cycle 0000 0001: 2 HCLK cycles …… 1111 1111: 256 HCLK cycles
15:8	WAIT	R/W	I/O x Wait Time Configure This bit takes HCLK as the clock cycle and defines the minimum hold time of the command. After the defined time, if the waiting signal is effective low, the hold time of the command will become longer. 0000 0000: Reserved 0000 0001: 2 HCLK cycles(+ wait cycle introduced by deasserting NWAIT) 0000 0010: 3 HCLK cycles …… 1111 1111: 256 HCLK cycles(+ wait cycle introduced by the Card deasserting NWAIT)

Field	Name	R/W	Description
23:16	HLD	R/W	I/O x Hold Time Configure This bit takes CLK as the clock cycle, and defines the hold time of address signal after sending the command. 0000 0000: Reserved 0000 0001: 1 HCLK cycle 0000 0010: 2 HCLK cycles …… 1111 1111: 255 HCLK cycles
31:24	HIZ	R/W	I/O x Databus Hiz Time Configure This bit takes HCLK as the clock cycle and defines the time of high-impedance state of data bus, which is only effective for write operation. 0000 0000: 0 HCLK cycle 0000 0001: 1 HCLK cycle …… 1111 1111: 255 HCLK cycles

4.7.2.6 ECC result register 2/3 (SMC_ECCRS2/3)

Offset address: 0x54 + 0x20 * (x-1), x=2 or 3

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	ECCRS	R	ECC result

4.8 DMC Introduction

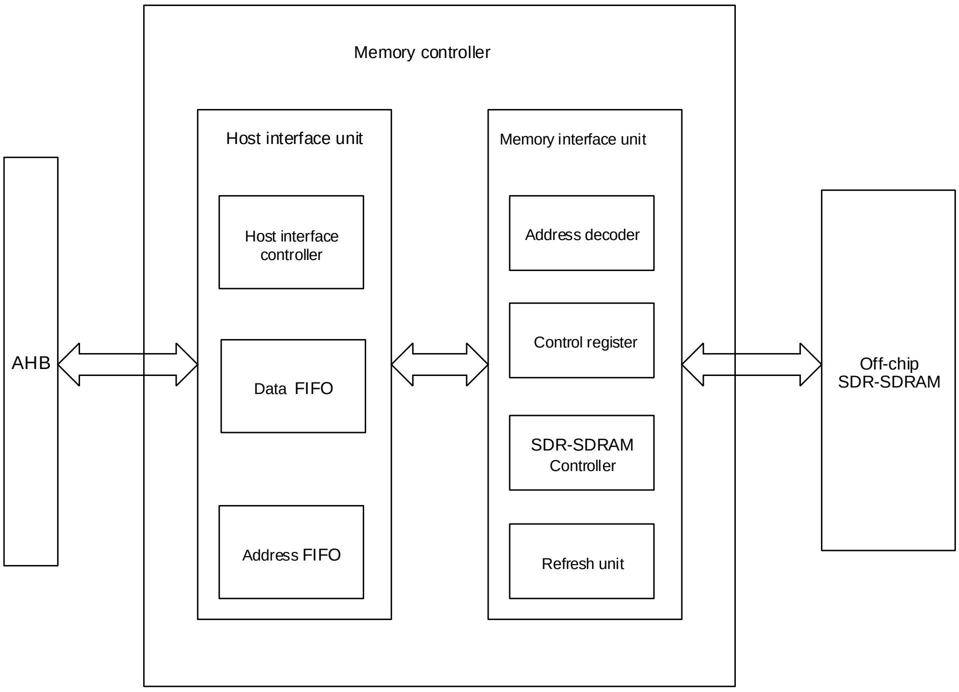
DMC is a dynamic memory controller, connected to off-chip SDR-SDRAM.

4.9 Main Characteristics of DMC

- 16-bit data width
- Up to 2MB off-chip SDR-SDRAM
- SDR-SDRAM timing and size are configurable
- SDR-SDRAM power-down mode supported
- SDR-SDRAM auto-refresh and self-refresh mode supported

4.10 DMC Structure Block Diagram

Figure 4 DMC Structure Block Diagram



4.11 DMC Functional Description

4.11.1 DMC External Memory Interface

The signal with the prefix "N" means low effective signal.

Table 24 DMC Pin (applicable to APM32E103xCxE non-sealed products)

Signal Name	Input/Output	Pin	Function
A0	Output	PF13	Address
A1	Output	PF14	Address
A2	Output	PF15	Address
A3	Output	PG0	Address
A4	Output	PE8	Address
A5	Output	PE9	Address
A6	Output	PE10	Address
A7	Output	PE11	Address
A8	Output	PE12	Address
A9	Output	PE13	Address
A10	Output	PF12	Address

Signal Name	Input/Output	Pin	Function
D0	Input/Output	PG12	Bidirectional data
D1	Input/Output	PG13	Bidirectional data
D2	Input/Output	PG14	Bidirectional data
D3	Input/Output	PG15	Bidirectional data
D4	Input/Output	PE3	Bidirectional data
D5	Input/Output	PE5	Bidirectional data
D6	Input/Output	PE6	Bidirectional data
D7	Input/Output	PF0	Bidirectional data
D8	Input/Output	PC10	Bidirectional data
D9	Input/Output	PC11	Bidirectional data
D10	Input/Output	PD2	Bidirectional data
D11	Input/Output	PD3	Bidirectional data
D12	Input/Output	PD4	Bidirectional data
D13	Input/Output	PD5	Bidirectional data
D14	Input/Output	PD6	Bidirectional data
D15	Input/Output	PG9	Bidirectional data
BA	Output	PF11	Bank address
CLK	Output	PE15	Clock
CKE	Output	PB11	Clock enable
LDQM	Input	PF10	16-bit data write
UNQM	Input	PB10	16-bit data read
NWE	Output	PF6	Write enable
NCAS	Output	PF5	Column address bit strobe command
NRAS	Output	PF4	Row address bit strobe command
NCS	Output	PF2	Chip selection

Table 25 DMC Pin (applicable to APM32E103VET6S)

信号名称	输入/输出	管脚	功能
A0	Output	PF13	Address
A1	Output	PF14	Address
A2	Output	PF15	Address
A3	Output	PG0	Address

信号名称	输入/输出	管脚	功能
A4	Output	PE8	Address
A5	Output	PE9	Address
A6	Output	PE10	Address
A7	Output	PE11	Address
A8	Output	PE12	Address
A9	Output	PE13	Address
A10	Output	PF12	Address
D0	Input/Output	PG12	Bidirectional data
D1	Input/Output	PG13	Bidirectional data
D2	Input/Output	PG14	Bidirectional data
D3	Input/Output	PG15	Bidirectional data
D4	Input/Output	PE3	Bidirectional data
D5	Input/Output	PE5	Bidirectional data
D6	Input/Output	PE6	Bidirectional data
D7	Input/Output	PF0	Bidirectional data
D8	Input/Output	PC10	Bidirectional data
D9	Input/Output	PC11	Bidirectional data
D10	Input/Output	PD2	Bidirectional data
D11	Input/Output	PD3	Bidirectional data
D12	Input/Output	PD4	Bidirectional data
D13	Input/Output	PD5	Bidirectional data
D14	Input/Output	PD6	Bidirectional data
D15	Input/Output	PG9	Bidirectional data
BA0	Output	PF11	Bank address
BA1	Output	PD12	Bank address
CKE	Output	PB11	Clock enable
CLK	Output	PE15	Clock
LDQM	Input	PF10	16-bit data write
UNQM	Input	PB10	16-bit data read
NWE	Output	PF6	Write enable
NCAS	Output	PF5	Column address bit strobe command

信号名称	输入/输出	管脚	功能
NRAS	Output	PF4	Row address bit strobe command
NCS	Output	PF2	Chip selection

4.11.2 DMC Configuration Example

Configure switch register DMC_SW as 1 to enable DMC;

The maximum value of SDRAM clock frequency is 50MHZ;

Configure the bit [29:28] of register RCM_CFG to decide the clock frequency of SDRAM;

Example: 512K×2BANKS×16BIT SDRAM

- Corresponding pin of row address: A0-A9, totally 11 bits
- Corresponding pin of column address: A0-A8, totally 8 bits
- Corresponding pin of Bank: BA0, 1 bit

The row address width bit of the configuration register DMC _CFG is set to 1010;

The column address width bit of the configuration register DMC_CFG is set to 0111;

The Bank address width bit of the configuration register DMC_CFG is set to 00.

4.12 DMC Register Address Mapping

Table 26 DMC Register Address Mapping

Register name	Description	Offset address
DMC_CFG	Configuration register	0x00
DMC_TIM0	Timing register 0	0x04
DMC_TIM1	Timing register 1	0x08
DMC_CTRL1	Control register 1	0x0C
DMC_REF	Refresh register	0x10
DMC_CHIPSEL	Chip selection register	0x14
DMC_MASK	Mask register	0x54
DMC_SW	Switch register	0x400
DMC_CTRL2	Control register 2	0x404

4.13 DMC Register Functional Description

4.13.1 Configuration register (DMC_CFG)

Offset address: 0x00

Reset value: 0x0014 1388

Field	Name	R/W	Description
2:0	Reserved		
4:3	BAWCFG	R/W	Bank Address Width Configure 00: The bank is 1 bit 01: The bank is 2 bits Others reserved
8:5	RAWCFG	R/W	Row Address Width Configure 0000-1001: Reserved 1010: The row address is 11 bits 1011: The row address is 12 bits …… 1111: The row address is 16 bits
12:9	CAWCFG	R/W	Column Address Width Configure 0000-0110: Reserved 0111: The column address is 8 bits 1000: The column address is 9 bits …… 1110: The column address is 15 bits 1111: Reserved
14:13	DWCFG	R/W	Data Width Configure 00: SDRAM data bit width is 16 bits Others reserved
31:15	Reserved		

4.13.2 Timing register 0 (DMC_TIM0)

Offset address: 0x04

Reset value: 0x019A 5252

Field	Name	R/W	Description
1:0	CASLSEL0	R/W	CAS Latency Select CAS = CASLSEL0+（ECASLSEL1<< 2） 00: 1 clock cycle 01: 2 clock cycles 10: 3 clock cycles 11: 4 clock cycles Others: Reserved

Field	Name	R/W	Description
5:2	RASMINTSEL	R/W	RAS Minimum Time Select These bits are used to select the minimum time between row activation and precharge. 0000: 1 clock cycle 0001: 2 clock cycles …… 1111: 16 clock cycles
8:6	DTIMSEL	R/W	RAS To CAS Delay Time Select 000: 1 clock cycle 001: 2 clock cycles …… 111: 8 clock cycles
11:9	PCPSEL	R/W	Precharge Period Select 000: 1 clock cycle 001: 2 clock cycles …… 111: 8 clock cycles
13:12	WRTIMSEL	R/W	Select Time Between The Last Data And The Next Precharge For Write 00: 1 clock cycle 01: 2 clock cycles 10: 3 clock cycles 11: 4 clock cycles
17:14	ARPSEL	R/W	Auto-Refresh Period Select These bits are used to define the minimum time interval between two auto-refresh commands. 0000: 1 clock cycle 0001: 2 clock cycles …… 1111: 16 clock cycles

21:18	XSR0	R/W	Minimum interval time from exiting self-refresh switch to activation command or auto-refresh read command
			XSR = XSR0 +（EXSR1 << 4）.
			XSR = 0 ~ 511 corresponds to 1 ~ 512 SDCLK clock cycles

25:22	ATACP	R/W	Active To Active Command Cycle 0000: 1 clock cycle 0001: 2 clock cycles …… 1111: 16 clock cycles
26	ECASLSEL1	R/W	Extended CAS Latency See TIM0_CASLSEL0 interpretation
31:27	EXSR1	R/W	Minimum interval time from exiting self-refresh switch to activation command or auto-refresh read command See TIM0_XSR0 interpretation

4.13.3 Timing register 1 (DMC_TIM1)

Offset address: 0x08

Reset value: 0x0007 4E20

Field	Name	R/W	Description
15:0	STBTIM	R/W	Stable Time After the SDRAM is powered on, how many clock cycles does it take to receive other commands.
19:16	ARNUMCFG	R/W	Number Of Auto-Refresh During Initialization Configure 0000: 1 auto-refresh 0001: 2 auto-refresh …… 1111: 16 auto-refresh
31:20	Reserved		

4.13.4 Control register 1 (DMC_CTRL1)

Offset address: 0x0C

Reset value: 0x0000 3048

Field	Name	R/W	Description
0	INIT	R/W	SDRAM Initialize 0: Invalid 1: Initialize SDRAM, and after SDRAM initialization, the hardware will automatically set 0 to this bit.
1	SRMEN	R/W	Put SDRM in Self-Refresh Mode Enable 0: Invalid 1: Enable
2	PDMEN	R/W	Put SDRM in Power-Down Mode Enable 0: Disable 1: Enable
3	PCACFG	R/W	Precharge Algorithm Configure 0: Precharge the row immediately after read operation. 1: Precharge the row after a period of delay upon completion of read operation.
4	FRBSREN	R/W	Full Refresh Before Entering Self-Refresh Mode 0: Only refresh one row before entering self-refresh mode 1: Refresh all rows before entering self-refresh mode
5	FRASREN	R/W	Full Refresh After Exit Self-Refresh Mode 0: Only refresh one row before after exiting self-refresh mode 1: Refresh all rows after exiting self-refresh mode
8:6	RDNUMCFG	R/W	Configure Number Of Registers Inserted In Read Data Path 000: 0 register 001: 1 register …… 111: 7 registers
9	MODESET	R/W	Mode Setup 1: Update SDRAM mode register. After the mode register is updated, the hardware will clear the bit automatically
10	Reserved		

Field	Name	R/W	Description
11	SRMFLG	R	Self-refresh mode flag 1: Indicate that the current SDRAM is stored in self-refresh mode. This bit is valid when FRBSREN position is 1.
15:12	BANKNUMCFG	R/W	Number Of Open Banks Configure The quantity of opened Banks in the SDRAM. 0000: 1 bank 0001: 2 banks …… 1111: 16 banks
31:16	Reserved		

4.13.5 Refresh register (DMC_REF)

Offset address: 0x10

Reset value: 0x0000 00C3

Field	Name	R/W	Description
15:0	RCYCCFG	R/W	Refresh Cycle Configure Specify the number of clock cycles between two consecutive refreshes.
31:16	Reserved		

4.13.6 Chip selection register (DMC_CHIPSEL)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	Reserved		
31:16	BACHIPSEL	R/W	Base Address For Chip Select The default is 0.

4.13.7 Mask register (DMC_MASK)

Offset address: 0x54

Reset value: 0x0000 000D

Field	Name	R/W	Description
4:0	MSIZESEL	R/W	Memory Size Select 0000: No memory connected 0001：64KB 0010：128KB 0011：256KB 0100：512KB 0101：1MB 0110：2MB 0111：4MB 1000：8MB 1001：16MB 1010：32MB

Field	Name	R/W	Description
			1011：64MB 1100：128MB 1101：256MB Others: Reserved
7:5	MTYPESEL	R/W	Memory Type Select 000：SDRAM Others: Reserved
31:8	Reserved		

4.13.8 Switch register (DMC_SW)

Offset address: 0x400

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	MCSW	R/W	Memory Controller Function Switch 0: Select SMC 1: Select DMC
31:1	Reserved		

4.13.9 Control register 2 (DMC_CTRL2)

Offset address: 0x404

Reset value: 0x0000 002E

Field	Name	R/W	Description
0	CPHACFG	R/W	Clock Phase Configure 0: The system clock is not reverse 1: The system clock displays reversely
1	RDDEN	R/W	RD Delay Function Enable 0: Enable 1: Disable
4:2	RDDCFG	R/W	RD Clock Delay Configure 000: 0 system clock 001: 1 system clock …… 111: 7 system clocks
5	WPEN	R/W	Write Pipe Enable 0: Disable 1: Enable
6	BUFFEN	R/W	DMC Accelerate Module Enable By opening the buffer area, wrap the burst data transmission, and improve the SDRAM read performance. Note: This bit is applicable only to APM32F103xC.
7	WRPBSEL	R/W	WRAP Burst Type Selection 0: Wrap4 burst transfer 1: Wrap8 burst transfer Note: This bit is applicable only to APM32F103xC.

Field	Name	R/W	Description
31:8	Reserved		

5 Reset and Clock Management (RCM)

5.1 Full Name and Abbreviation Description of Terms

Table 27 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Reset and Clock Management	RCM
Reset	RST
Power-On Reset	POR
Power-Down Reset	PDR
High Speed External Clock	HSECLK
Low Speed External Clock	LSECLK
High Speed Internal Clock	HSICLK
Low Speed Internal Clock	LSICLK
Phase Locked Loop	PLL
Main Clock Output	MCO
Calibrate	CAL
Trim	TRIM
Wakeup	WUP
Automatic Wakeup	AWUP
Backup	BAKP
Low Power	LPWR
Clock Security System	CSS
Non Maskable Interrupt	NMI

5.2 Reset Functional Description

The supported reset is divided into three forms, namely, system reset, power reset and backup area reset.

5.2.1 System Reset

5.2.1.1 "System reset" reset source

The reset source is divided into external reset source and internal reset source.

External reset source:

- Low level on NRST pin.

Internal reset source:

- Window watchdog termination count (WWDT reset)
- Independent watchdog termination count (IWDT reset)
- Software reset (SW reset)
- Low-power management reset
- Power reset

A system reset will occur in case of any of the above events. Besides, the reset event source can be identified by viewing the reset flag bit in RCM_CSTS (control/state register).

Generally, when the system is reset, all registers except the registers in RCM_CSTS (control/state register) reset flag bit and backup area will be reset to the reset state.

Software reset

Software can be reset by putting SYSRESETREQ in Arm® Cortex®-M3 interrupt application and reset control register to "1".

Low-power management reset

Low-power management may reset in two cases, one is when entering the standby mode, and the other is when entering the stop mode. In these two cases, if RSTSTDB bit (in standby mode) or RSTSTOP bit (in stop mode) in user selection byte is cleared, the system will be reset rather than entering the standby or stop mode.

For more information about user option bytes, refer to "Flash memory".

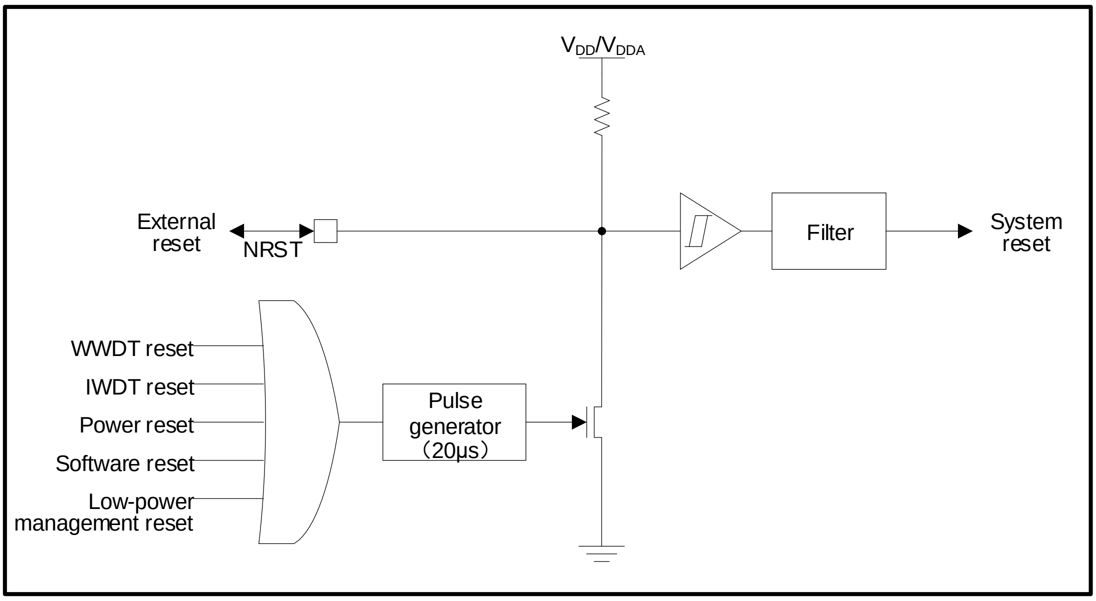
5.2.1.2 "System Reset" reset circuit

The reset source is used in the NRST pin, which remains low in reset process.

The internal reset source generates a delay of at least 20μs pulse on the NRST pin through the pulse generator, which causes the NRST to maintain the level to generate reset; the external reset source directly pulls down the NRST pin level to generate reset.

The "system reset" reset circuit is shown in the figure below.

Figure 5 "System Reset" Reset Circuit



5.2.2 Power Reset

"Power reset" reset source

"Power reset" reset source is as follows:

- Power-on reset (POR reset)
- Power-down reset (PDR reset)
- Wake up from standby mode

A power reset will occur in case of any of the above events.

Power reset will reset all registers except that in backup area.

5.2.3 Backup Domain Reset

"Backup domain reset" reset source

"Backup domain reset" reset source is as follows:

- Software resets and sets the BDRST bit in RCM_BDCTRL (backup domain control register)
- V_{DD} or V_{BAT} is powered on when V_{DD} or V_{BAT} is powered down

A backup domain reset will occur in case of any of the above events.

The backup area reset has two special resets, which only affect backup area.

5.3 Functional Description of Clock Management

Clock sources of the whole system are: HSECLK, LSECLK, HSICLK, LSICLK, PLL. For the characteristics of the clock source, please refer to the relevant chapter of "Electrical Characteristics" in the data manual.

5.3.1 External Clock Source

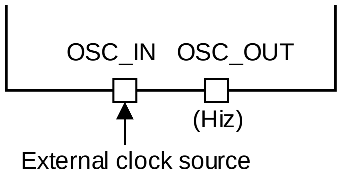
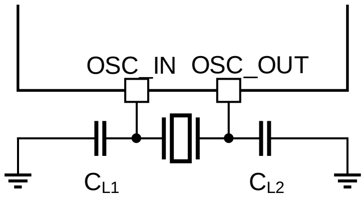
The external clock signal includes HSECLK (high-speed external clock signal) and LSECLK (low-speed external clock signal).

There are two kinds of external clock sources:

- External crystal/ceramic resonator
- External clock of user

The hardware configuration of the two kinds of clock sources is shown in the figure below.

Figure 6 HSECLK/LSECLK Clock Source Hardware Configuration

Clock source	Hardware configuration
External clock	
Crystal/Ceramic resonator	

In order to reduce the distortion of clock output and shorten the start-up stabilization time, the crystal/ceramic resonator and load capacitor must be as close to the oscillator pin as possible. The value of load capacitance (C_{L1}, C_{L2}) must be adjusted according to the selected oscillator.

5.3.1.1 HSECLK high-speed external clock signal

HSECLK clock signal is generated by HSECLK external crystal/ceramic resonator and HSECLK external clock two kinds of clock sources.

Table28 Clock Source Generting HSECLK

Name	**Instruction**
External clock source (HSECLK bypass)	Provide clock to MCU through OSC_IN pin. The signal can be generated by ordinary function signal transmitter (in debugging), crystal oscillator and other signal generators; the waveform can be square wave, sine wave or triangle wave with 50% duty cycle, and the maximum frequency is up to 25MHz. For hardware connection, it must be connected to OSC_IN pin, ensuring OSC_OUT pin is suspended; for MCU configuration, the user can select this mode by setting HSEBCFG and HSEEN bits in RCM_CTRL.

Name	Instruction
External crystal/ceramic resonator (HSECLK crystal)	The clock is provided to MCU by the resonator, and the resonator includes crystal resonator and ceramic resonator. The frequency range is 4-16MHz. When needing to connect OSC_IN and OSC_OUT to the resonator , start and close by setting the HSEEN bit in RCM_CTRL. HSERDYFLG bit in clock control register RCM_CTRL is used to indicate whether the high-speed external oscillator is stable. After startup, the clock is not released until this bit is set to "1" by hardware. If interrupt is allowed in RCM_INT (clock interrupt register), corresponding interrupt will be generated.

5.3.1.2 LSECLK low-speed external clock signal

LSECLK clock signal is generated by LSECLK external crystal/ceramic resonator and LSECLK external clock two kinds of clock sources.

Table 29 Clock Source Generting LSECLK

Name	Instruction
External clock source (LSECLK bypass)	The cock is provided to to MCU through OSC32_IN pin. The signal can be generated by ordinary function signal transmitter (in debugging), crystal oscillator and other signal generators; the waveform can be square wave, sine wave or triangle wave with 50% duty cycle, and the signal frequency needs to be 32.768kHz. For hardware connection, it must be connected to OSC32_IN pin, ensuring OSC32_OUT pin is suspended; for MCU configuration, the user can select this mode by setting LSEBCFG and LSEEN bits in RCM_BDCTRL.
External crystal/ceramic resonator (LSECLK crystal)	The clock is provided to MCU by the resonator, and the resonator includes crystal resonator and ceramic resonator. The frequency is 32.768kHz. OSC32_IN、OSC32_OUT needs to be connected to the oscillator which can be enabled and disabled through LSEEN bit in RCM_BDCTRL. LSERDYFLG in RCM_BDCTRL indicates whether LSECLK crystal oscillator is stable. At startup stage, LSECLK clock signal is not released until this bit is set to "1" by hardware. If it is allowed in the clock interrupt register, an interrupt request can be generated.

5.3.2 Internal Clock Source

The internal clock includes HSICLK (high-speed internal clock signal) and LSICLK (low-speed internal clock signal).

5.3.2.1 HSICLK high-speed internal clock signal

HSICLK clock signal is generated by internal 8MHz RC oscillator.

The RC oscillator frequency of different chips is different, and that of the same chip may be different with the change of temperature and voltage; the HSICLK clock frequency of each chip has been calibrated to 1% (25 ℃, V_{DD}=V_{DDA}=3.3V) by the manufacturer before leaving the factory. When the system is reset, the value calibrated by the manufacturer will be loaded to

RCM_CTRL; in addition, the users can further adjust the frequency by setting HSICLKTRM in RCM_CTRL according to the application environment (temperature and voltage) of the site.

HSIRDYFLG bit can be used to indicate whether HSICLK RC oscillator is stable. In the clock startup process, HSICLK RC output clock is not released until the HSIRDYFLG bit is set to 1 by hardware. HSICLK RC can be started or closed by HSIEN bit in RCM_CTRL.

Compared with HSECLK crystal oscillator, RC oscillator can provide system clock without any external device; the start time of RC oscillator is shorter than that of HSECLK crystal oscillator; even after calibration, its clock frequency accuracy is still inferior to that of HSECLK crystal oscillator.

5.3.2.2 LSICLK low-speed internal clock signal

Main characteristics of LSICLK

LSICLK is generated by RC oscillator. The frequency may change along with the change of temperature and voltage. It can keep running in stop and standby mode and provide clock for independent watchdog and automatic wake-up unit.

LSICLK can be started or closed by LSIEN bit in RCM_CSTS. LSIRDYFLG bit in RCM_CSTS indicates whether the low-speed internal oscillator is stable. At startup stage, the clock is not released until this bit is set to "1" by hardware. If allowed in RCM_INT, LSICLK interrupt application will be generated.

LSICLK calibration

Note: LSICLK calibration can be performed only for large-capacity products with TMR5

The purpose of calibrating LSICLK oscillator is to compensate its frequency offset. After calibration, it can get RTC clock base with certain precision and IWDT timeout period.

First, set the channel 4 of TMR5 to the input capture mode, connect the LSICLK clock and measure its clock frequency. Then set the 20-bit prescaler of RTC according to the required RTC time base and the IWDT timeout period with the HSECLK clock frequency as the accuracy guarantee.

5.3.3 PLL (Phase Locked Loop)

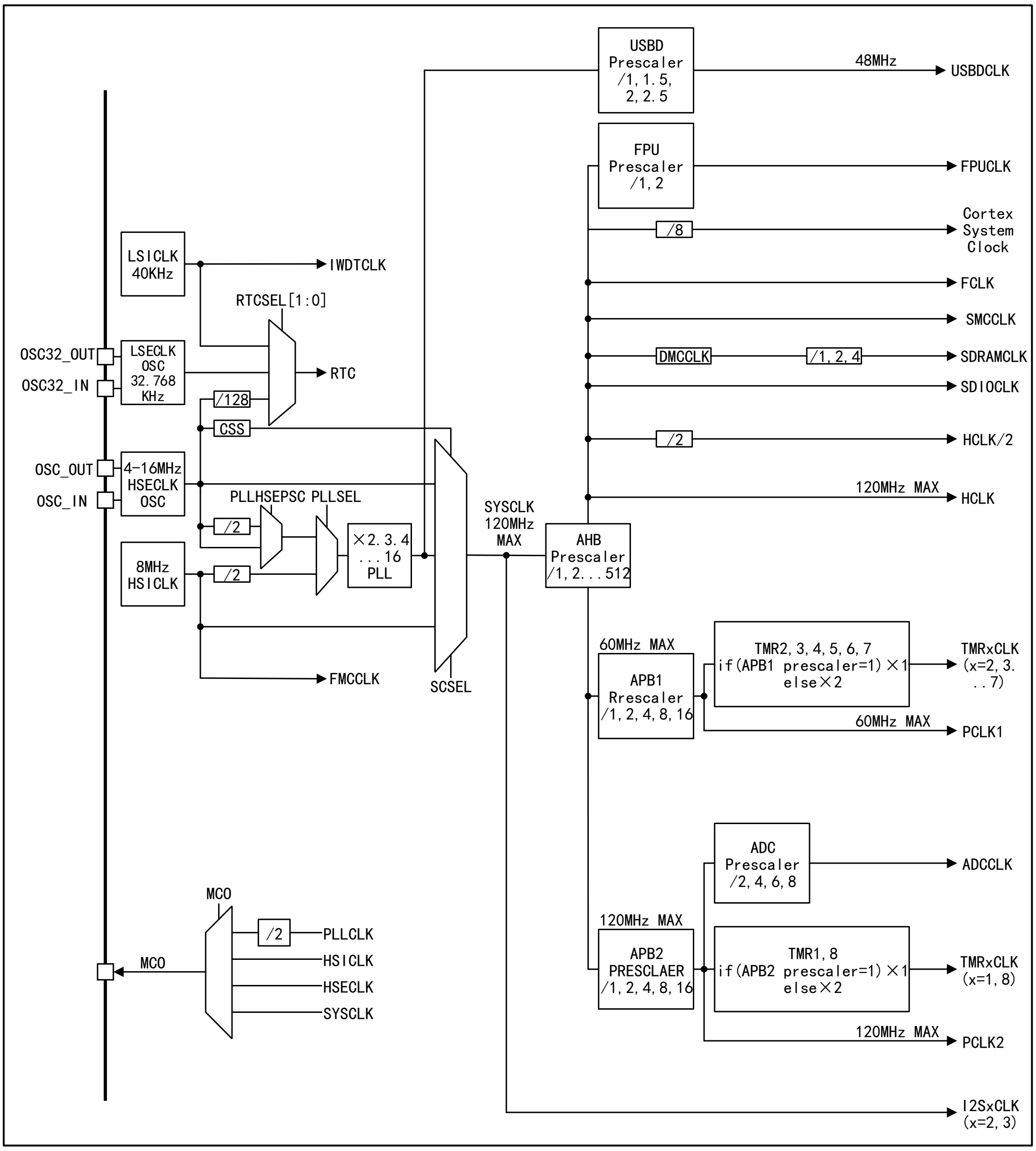
The internal PLL can be used to double the frequency of HSICLK output clock or HSECLK crystal output clock.

HSICLK/2 or HSECLK can be selected as PLL input clock source, and output PLLCLK after PLL frequency multiplication (frequency multiplication factor can be selected). The clock source and multiplication factor should be selected before being activated. Once PLL is activated, the selection cannot be changed.

When PLL is ready and PLL interrupt in RCM_INT is allowed, PLL can send interrupt request.

5.3.4 Clock Tree

Figure 7 APM32E103xCxE Clock Tree



Note:

(1) HCLK means AHB clock.

(2) PCLK1 and PCLK2 are clock signal connected to the peripheral of APB1 and APB2 respectively.

(3) FCLK is running clock of Arm® Cortex®-M3.

(4) The frequency of AHB, APB2 (high-speed APB) and APB1 (low-speed APB) domains can be configured through multiple prescalers. Besides, the maximum frequency of AHB and APB2 domain is 120MHz, while the maximum allowable frequency of APB1 is 60MHz.

(5) When USBD is used, PLL must be set to output 48/72/96/120MHz clock, which is used to provide 48MHz USBDCLK clock after frequency division by USBD frequency divider.

(6) When needing to run the peripheral connected to AHB and APB, it is required to turn on the corresponding enable end to make the peripheral get the clock signal.

(7) SysTick (system timer) can be provided by the clock signal after frequency division of HCLK8. Different clock sources can be selected by setting SysTick control and status register.

(8) Frequency assignment of all TMRxCLK (timer clocks) is automatically set by the hardware according to the following two situations:

- If the corresponding APB prescaler factor is 1, the clock frequency of the timer is the same as that of the APB bus.
- Otherwise, the clock frequency of the timer will be set to twice the frequency of the APB bus connected to it.

(9) Moreover, the frequency of TMRx (x=2,3,4,5,6,7) clock signals is divided by APB1, and the frequency of TMRx (x=1,8) clock signals is dividied by APB2.

(10) Only APM32E103VET6S products have sealed SDRAM, but no SDIO.

5.3.5 Clock Source Selection of RTC

HSECLK/128, LSECLK or LSICLK can be selected as RTCCLK clock source by setting RTCSRCSEL bit in RCM_BDCTRL. The selection of clock source can be changed only when the backup domain is reset.

Because LSECLK is in the backup domain, and HSECLK and LSICLK are not in the backup domain, different clocks will be selected as the clock source; the working condition of RTCs are different, and see the following table for details:

Table 30 Working Condition of RTC When RTC Selects Different Clock Sources

Clock source	Working condition
LSECLK is selected as RTC clock	As long as V_{BAT} maintains power supply, RTC will continue to work even if V_{DD} is powered off
LSICLK is selected as automatic wake-up unit	If V_{DD} is powered off, AWUP state cannot be guaranteed.
HSECLK/128 as RTC clock	If the V_{DD} is powered off or the internal voltage regulator is turned off (the power supply of 1.3V domain is cut off), the RTC state is uncertain, so the BPWEN bit (cancel the write protection of backup area) of PMU_CTRL (power control register) must be set to "1".

5.3.6 Clock Source Selection of IWDT

When IWDT (independent watchdog) is opened, LSICLK oscillator will be opened by force, and when it is stable, the clock signal will be provided to IWDT. After LSICLK is opened by force, it will always be open and cannot be closed.

5.3.7 Clock Source Selection of MCO

When the corresponding GPIO port register is configured with corresponding function, the clock signal can be selected to be output to MCO pin by MCOSEL in configuration register RCM_CFG (clock configuration register). See the clock tree or MCOSEL bit instructions for specific clock signal.

5.3.8 Clock Source Selection of SYSCLK

After system reset, HSICLK oscillator is selected as the system clock, which cannot be stopped. If you want to switch the SYSCLK clock source, you must wait until the destination clock source is ready (i.e. the destination clock source is stable). The target clock source can be HSECLK and PLLCLK, and the clock source of PLLCLK can be HSECLK and HSICLK/2.

The state bit of RCM_CFG can indicate the ready clock and selected SYSCLK clock source.

5.3.9 CSS Clock Security System

In order to prevent MCU from normal operation due to external crystal oscillator short circuit, MCU can activate CSS clock security system through software. After the security system is activated, if the HSECLK oscillator is used as the system clock directly or indirectly (used as the PLL input clock and PLL is used as the system clock), the external HSECLK oscillator will be turned off when the HSECLK clock fails, and the system clock will automatically switch to HSICLK. At this time, the PLL which selects HSECLK as the clock input and as the system clock input source will also be turned off.

Note: When CSS is activated by software and HSECLK clock fails, CSS interrupt and NMI (non-maskable interrupt) will be generated. Since NMI is executed continuously before CSS interrupt is cleared, CSSIF bit in RCM_INT register shall be set to clear the interrupt.

5.4 Register Address Mapping

Table 31 RCM Register Address Mapping

Register name	Description	Offset address
RCM_CTRL	Clock control register	0x00
RCM_CFG	Clock configuration register	0x04
RCM_INT	Clock interrupt register	0x08
RCM_APB2RST	APB2 peripheral reset register	0x0C
RCM_APB1RST	APB1 peripheral reset register	0x10
RCM_AHBCLKEN	AHB peripheral clock enable register	0x14
RCM_APB2CLKEN	APB2 peripheral clock enable register	0x18
RCM_APB1CLKEN	APB1 peripheral clock enable register	0x1C

Register name	Description	Offset address
RCM_BDCTRL	Backup domain control register	0x20
RCM_CSTS	Control/State register	0x24

5.5 Register Functional Description

5.5.1 Clock control register (RCM_CTRL)

Offset address: 0x00

Reset value: 0x0000 XX83; X means undefined

Access: Access in the form of word, half word and byte, without wait cycle

Field	Name	R/W	Description
0	HSIEN	R/W	High Speed Internal Clock Enable Set 1 or clear 0 by software. HSICLK is an RC oscillator. When one of the following conditions occurs, it will be set to 1 by the hardware: power-on start, software reset, wake-up from standby mode, wake-up from stop mode, failure of external high-speed clock source (as system clock or providing system clock through PLL). When HSICLK is used as system clock or provides system clock through PLL, this bit cannot be cleared. 0: HSICLK RC oscillator is disabled 1: HSICLK RC oscillator is turned on
1	HSIRDYFLG	R	High Speed Internal Clock Ready Flag 0: HSICLK RC oscillator is not stable 1: HSICLK RC oscillator is stable
2	Reserved		
7:3	HSITRM	R/W	High Speed Internal Clock Trim The product has been calibrated to 8MHz±1% when leaving the factory. However, it changes as the temperature and voltage changes, but the frequency of HSICLK RC oscillator can be adjusted by HSITRM.
15:8	HSICAL	R	High Speed Internal Clock Calibrate It will be calibrated to 8MHz±1% before leaving the factory. When the system is started up, the calibration parameters will be automatically written to the register.
16	HSEEN	R/W	High Speed External Clock Enable When entering the standby or stop mode, this bit is cleared by hardware and HSECLK is turned off; when HSECLK is used as system clock source or the system clock is provided through PLL, this bit cannot be cleared. 0: HSECLK is disabled 1: HSECLK is enabled
17	HSERDYFLG	R	High Speed External Clock Ready Flag When HSECLK is stable, this bit is set to 1 by hardware and cleared by software. 0: HSECLK is not stable 1: HSECLK is stable

Field	Name	R/W	Description
18	HSEBCFG	R/W	High Speed External Clock Bypass Configure Bypass mode refers to the mode in which external clock is used as the HSECLK clock source; otherwise the resonator is used as the HSECLK clock source. 0: Non-bypass mode 1: Bypass mode
19	CSSEN	R/W	Clock Security System Enable 0: Disable 1: Enable
23:20	Reserved		
24	PLLEN	R/W	PLL Enable When entering the standby and stop mode, this bit is cleared by the hardware; when PLLCLK has been configured (or in the process of configuration) as the clock source of the system clock, this bit cannot be cleared; in other cases, it can be set to 1 or cleared by the software. 0: PLL is disabled 1: PLL is enabled
25	PLLRDYFLG	R	PLL Clock Ready Flag PLL is set to 1 by hardware after it is locked. 0: PLL is unlocked 1: PLL is locked
31:26	Reserved		

5.5.2 Clock configuration register (RCM_CFG)

Offset address: 0x04

Reset value: 0x0000 0000

All bits of this register are set or cleared by software.

Access: Access in the form of word, half word and byte, with 0 to 2 wait cycles. 1 or 2 wait cycles are inserted only when the access occurs during clock switching.

Field	Name	R/W	Description
1:0	SCLKSEL	R/W	System Clock Source Select Select system clock source. When returning from stop or standby mode or the HSE directly or indirectly used as system clock fails, the hardware selects HSICLK as system clock by force (if the clock security system has been started) 00: HSICLK is used as system clock 01: HSECLK is used as system clock 10: PLLCLK is used as system clock 11: Unavailable
3:2	SCLKSELSTS	R	System Clock Selection Status Indicate which clock source is used as system clock. 00: HSICLK is used as system clock 01: HSECLK is used as system clock

Field	Name	R/W	Description
			10: PLLCLK output is used as system clock 11: Unavailable
7:4	AHBPSC	R/W	AHB Clock Prescaler Factor Configure Control the prescaler factor of AHB clock. 0xxx: No frequency division for SYSCLK 1000: SYSCLK 2-divided frequency 1001: SYSCLK 4-divided frequency 1010: SYSCLK 8-divided frequency 1011: SYSCLK 16-divided frequency 1100: SYSCLK 64-divided frequency 1101: SYSCLK 128-divided frequency 1110: SYSCLK 256-divided frequency 1111: SYSCLK 512-divided frequency Note: When the prescaler factor of AHB clock is greater than 1, the prefetch buffer must be enabled.
10:8	APB1PSC	R/W	APB1 Clock Prescaler Factor Configure Control the prescaler factor of low-speed APB1 clock (PCLK1) Warning: Softwre must ensure APB1 clock frequency is not greater than 36MHz. 0xx: No frequency division for HCLK 100: HCLK 2-divided frequency 101: HCLK 4-divided frequency 110: HCLK 8-divided frequency 111: HCLK 16-divided frequency
13:11	APB2PSC	R/W	APB1 Clock Prescaler Factor Control the prescaler factor of high-speed APB2 clock (PCLK2) 0xx: No frequency division for HCLK 100: HCLK 2-divided frequency 101: HCLK 4-divided frequency 110: HCLK 8-divided frequency 111: HCLK 16-divided frequency
15:14	ADCPSC	R/W	ADC Clock Prescaler Factor Configure Determine ADC clock frequency 00: PCLK2 is used as ADCCLK after two divided frequency 01: PCLK2 is used as ADCCLK after four divided frequency 10: PCLK2 is used as ADCCLK after six divided frequency 11: PCLK2 is used as ADCCLK after eight divided frequency
16	PLLSRCSEL	R/W	PLL Clock Source Select Select PLL input clock source. 0: HSICLK RC oscillator clock is used as PLL input clock after 2 divided frequency 1: HSECLK is used as PLL input clock Note: This bit can be written only when PLL is disabled.
17	PLLHSEPSC	R/W	HSECLK Prescaler Factor for PLL Clock Source Used as PLL input clock after HSECLK frequency division. 0: No frequency division for HSECLK 1: HSECLK2 frequency division

Field	Name	R/W	Description
			Note: This bit can be written only when PLL is disabled.
21:18	PLLMULCFG	R/W	PLL Multiplication Factor Configure Determine PLL multiplication factor. This bit can be written only when PLL is disabled. 0000: PLLCLK 2-multiple frequency output 0001: PLLCLK 3-multiple frequency output 0010: PLLCLK 4-multiple frequency output 0011: PLLCLK 5-multiple frequency output 0100: PLLCLK 6-multiple frequency output 0101: PLLCLK 7-multiple frequency output 0110: PLLCLK 8-multiple frequency output 0111: PLLCLK 9-multiple frequency output 1000: PLLCLK 10-multiple frequency output 1001: PLLCLK 11-multiple frequency output 1010: PLLCLK 12-multiple frequency output 1011: PLLCLK 13-multiple frequency output 1100: PLLCLK 14-multiple frequency output 1101: PLLCLK 15-multiple frequency output 1110: PLLCLK 16-multiple frequency output 1111: PLLCLK 16-multiple frequency output Note: The output frequency of PLLCLK cannot be greater than 96MHz.
23:22	USBDPSC	R/W	USBD Prescaler Factor Configure USBD clock generting 48MHz. Before enabling USBD clock in RCM_APB1CLKEN register, it must be ensured that this bit has been effective. If USBD1/2 clock is enabled, this bit cannot be cleared. 0: PLLCLK is used as USBD clock after 1.5 divided frequency 1: PLLCLK is directly used as USBD clock 2: PLLCLK is used as USBD clock after 2 divided frequency 3: PLLCLK is used as USBD clock after 2.5 divided frequency
26:24	MCOSEL	R/W	Main Clock Output Select Set 1 or clear 0 by software. 0xx: No clock output 100: System clock (SYSCLK) output 101: Internal RC oscillator clock (HSI) output 110: External oscillator clock (HSECLK) output 111: PLLCLK is output after two divided frequency Note: 1. The clock output may be truncated when starting and switching the MCO clock source. 2. When the system clock is output to the MCO pin, please ensure that the output clock frequency is not greater than 50MHz (maximum frequency of I/O port).
27	FPUPSC	R/W	FPU Clock Prescaler Factor Configure 0: HCLK is used as FPU clock 1: HCLK is used as FPU clock after 2 divided frequency
29:28	SDRAMPSC	R/W	SDRAM Clock Prescaler Configure

Field	Name	R/W	Description
			00: DMC clock is used as the SDRAM clock 01: DMC clock is used as SDRAM clock after 2 divided frequency Others: DMC clock is used as SDRAM clock after 4 divided frequency Note: The maximum clock frequency of SDRAM is 50MHz.
31:30	Reserved		

5.5.3 Clock interrupt register (RCM_INT)

Offset address: 0x08

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle.

Field	Name	R/W	Description
0	LSIRDYFLG	R	LSICLK Ready Interrupt Flag When the internal low-speed clock is ready and the LSIRDYEN bit is set to 1, it is set to 1 by the hardware. The software clears the bit by setting 1LSIRDYCLR. 0: No LSICLK ready interrupt 1: LSICLK ready interrupt occurred
1	LSERDYFLG	R	LSECLK Ready Interrupt Flag When the external low-speed clock is ready and the LSERDYEN bit is set to 1, it is set to 1 by the hardware. The software clears the bit by setting 1LSERDYCLR. 0: No LSECLK ready interrupt 1: LSECLK ready interrupt occurred
2	HSIRDYFLG	R	HSICLK Ready Interrupt Flag When the internal high-speed clock is ready and the HSIRDYEN bit is set to 1, it is set to 1 by the hardware. The software clears the bit by setting 1HSIRDYCLR. 0: No HSICLK ready interrupt 1: HSICLK ready interrupt occurred
3	HSERDYFLG	R	HSECLK Ready Interrupt Flag When the external low-speed clock is ready and the HSERDYCLR bit is set to 1, it is set to 1 by the hardware. The software clears the bit by setting 1 HSERDYCLR. 0: No HSECLK ready interrupt 1: HSECLK ready interrupt occurred
4	PLLRDYFLG	R	PLL Ready Interrupt Flag When PLL is ready and PLLRDYEN bit is set to 1, it is set to 1 by the hardware. The software clears the bit by setting 1 PLLRDYCLR. 0: No clock ready interrupt caused by PLL locked 1: Clock ready interrupt caused by PLL locked
6:5	Reserved		

Field	Name	R/W	Description
7	CSSFLG	R	Clock Security System Interrupt Flag When the external 4-16MHz oscillator clock fails, it is set to 1 by hardware. The software clears the bit by setting CSSCLR bit. 0: No security system interrupt caused by HSECLK failure 1: Clock security system interrupt is caused by HSECLK failure
8	LSIRDYEN	R/W	LSICLK Ready Interrupt Enable Enable or disable internal 40kHz RC oscillator ready interrupt. 0: Disable 1: Enable
9	LSERDYEN	R/W	LSECLK Ready Interrupt Enable Enable external 32kHz RC oscillator ready interrupt. 0: Disable 1: Enable
10	HSIRDYEN	R/W	HSICLK Ready Interrupt Enable Enable the internal 8MHz RC oscillator ready interrupt. 0: Disable 1: Enable
11	HSERDYEN	R/W	HSECLK Ready Interrupt Enable Enable external 4-16MHz oscillator ready interrupt. 0: Disable 1: Enable
12	PLLRDYEN	R/W	PLL Ready Interrupt Enable Enable PLL ready interrupt. 0: Disable 1: Enable
15:13	Reserved		
16	LSIRDYCLR	W	LSICLK Ready Interrupt Clear Clear LSICLK ready interrupt flag bit LSIRDYFLG. 0: No effect 1: Clear
17	LSERDYCLR	W	LSECLK Ready Interrupt Clear Clear LSECLK ready interrupt flag bit LSERDYFLG. 0: No effect 1: Clear
18	HSIRDYCLR	W	HSICLK Ready Interrupt Clear Clear HSICLK ready interrupt flag bit HSIRDYFLG. 0: No effect 1: Clear
19	HSERDYCLR	W	HSECLK Ready Interrupt Clear Clear HSECLK ready interrupt flag bit HSERDYFLG. 0: No effect 1: Clear

Field	Name	R/W	Description
20	PLLRDYCLR	W	PLL Ready Interrupt Clear Clear PLL ready interrupt flag bit PLLRDYFLG. 0: No effect 1: Clear
22:21	Reserved		
23	CSSCLR	W	Clock Security System Interrupt Clear Clear the security system interrupt flag bit CSSFLG. 0: No effect 1: Clear
31:24	Reserved		

5.5.4 APB2 peripheral reset register (RCM_APB2RST)

Offset address: 0x0C

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle.

All bits can be reset or cleared by software.

Field	Name	R/W	Description
0	AFIO	R/W	Alternate Function I/O Reset 0: No effect 1: Reset
1	Reserved		
2	PA	R/W	IO Port A Reset 0: No effect 1: Reset
3	PB	R/W	IO Port B Reset 0: No effect 1: Reset
4	PC	R/W	IO Port C Reset 0: No effect 1: Reset
5	PD	R/W	IO Port D Reset 0: No effect 1: Reset
6	PE	R/W	IO Port E Reset 0: No effect 1: Reset
7	PF	R/W	IO Port F Reset 0: No effect 1: Reset
8	PG	R/W	IO Port G Reset 0: No effect 1: Reset

Field	Name	R/W	Description
9	ADC1	R/W	ADC1 Reset 0: No effect 1: Reset ADC1
10	ADC2	R/W	ADC2 Reset 0: No effect 1: Reset
11	TMR1	R/W	TMR1 Timer Reset 0: No effect 1: Reset
12	SPI1	R/W	SPI1 Reset 0: No effect 1: Reset
13	TMR8	R/W	TMR8 Timer Reset 0: No effect 1: Reset
14	USART1	R/W	USART1 Reset 0: No effect 1: Reset
15	ADC3	R/W	ADC3 Reset 0: No effect 1: Reset
31:16	Reserved		

5.5.5 APB1 peripheral reset register (RCM_APB1RST)

Offset address: 0x10

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte, without wait cycle

Field	Name	R/W	Description
0	TMR2	R/W	Timer 2 Reset 0: No effect 1: Reset
1	TMR3	R/W	Timer 3 Reset 0: No effect 1: Reset
2	TMR4	R/W	Timer 4 Reset Set 1 or clear 0 by software 0: No effect 1: Reset
3	TMR5	R/W	Timer5 Reset 0: No effect 1: Reset
4	TMR6	R/W	Timer 6 Reset 0: No effect 1: Reset

Field	Name	R/W	Description
5	TMR7	R/W	Timer 7 Reset 0: No effect 1: Reset
10:6	Reserved		
11	WWDT	R/W	Window Watchdog Reset 0: No effect 1: Reset
13:12	Reserved		
14	SPI2	R/W	SPI2 Reset 0: No effect 1: Reset
15	SPI3	R/W	SPI3 Reset 0: No effect 1: Reset
16	Reserved		
17	USART2	R/W	USART2 Reset 0: No effect 1: Reset
18	USART3	R/W	USART3 Reset 0: No effect 1: Reset
19	UART4	R/W	UART4 Reset 0: No effect 1: Reset
20	UART5	R/W	UART5 Reset 0: No effect 1: Reset
21	I2C1	R/W	I2C1 Reset 0: No effect 1: Reset
22	I2C2	R/W	I2C2 Reset 0: No effect 1: Reset
23	USBD	R/W	USBD1/2 Reset 0: No effect 1: Reset
24	Reserved		
25	CAN1	R/W	CAN1 Reset 0: No effect 1: Reset
26	CAN2	R/W	CAN2 Reset 0: No effect 1: Reset

Field	Name	R/W	Description
27	BAKP	R/W	Backup Interface Reset 0: No effect 1: Reset
28	PMU	R/W	Power Interface Reset 0: No effect 1: Reset
29	DAC	R/W	DAC Reset 0: No effect 1: Reset
31:30	Reserved		

5.5.6 AHB peripheral clock enable register (RCM_AHBCLKEN)

Offset address: 0x14

Reset value: 0x0000 0014

Access: Access in the form of word, half word and byte, without wait cycle

All bits can be reset or cleared by software.

Note: When the peripheral clock is not enabled, the software cannot read the value of the peripheral register, and the value returned is always 0x0.

Field	Name	R/W	Description
0	DMA1	R/W	DMA1 Clock Enable 0: Disable 1: Enable
1	DMA2	R/W	DMA2 Clock Enable 0: Disable 1: Enable
2	SRAM	R/W	SRAM Interface Clock Enable Enable SRAM clock in sleep mode. 0: Disable 1: Enable
3	FPU	R/W	FPU Clock Enable 0: Disable 1: Enable
4	FMC	R/W	FMC Clock Enable Enable the flash interface circuit clock in sleep mode. 0: Disable 1: Enable
5	Reserved		
6	CRC	R/W	CRC Clock Enable 0: Disable 1: Enable
7	Reserved		
8	EMMC	R/W	EMMC Clock Enable 0: Disable 1: Enable

Field	Name	R/W	Description
9	Reserved		
10	SDIO	R/W	SDIO Clock Enable 0: Disable 1: Enable
31:11	Reserved		

5.5.7 APB2 peripheral clock enable register (RCM_APB2CLKEN)

Offset address: 0x18

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte

Usually there is no wait cycle. However, when the peripheral on the APB2 bus is accessed, the waiting state will be inserted until the APB2 peripheral access ends.

All bits can be reset or cleared by software.

Note: When the peripheral clock is not enabled, the software cannot read the value of the peripheral register, and the value returned is always 0x0.

Field	Name	R/W	Description
0	AFIO	R/W	Alternate Function I/O Clock Enable 0: Disable 1: Enable
1	Reserved		
2	PA	R/W	I/O Port A Clock Enable 0: Disable 1: Enable
3	PB	R/W	I/O Port B Clock Enable 0: Disable 1: Enable
4	PC	R/W	I/O Port C Clock Enable 0: Disable 1: Enable
5	PD	R/W	I/O Port D Clock Enable 0: Disable 1: Enable
6	PE	R/W	I/O Port E Clock Enable 0: Disable 1: Enable
7	PF	R/W	I/O Port F Clock Enable 0: Disable 1: Enable
8	PG	R/W	I/O Port G Clock Enable 0: Disable 1: Enable

Field	Name	R/W	Description
9	ADC1	R/W	ADC 1 Interface Clock Enable 0: Disable 1: Enable
10	ADC2	R/W	ADC 2 Interface Clock Enable 0: Disable 1: Enable
11	TMR1	R/W	TMR1 Timer Clock Enable 0: Disable 1: Enable
12	SPI1	R/W	SPI 1 Clock Enable 0: Disable 1: Enable
13	TMR8	R/W	TMR8 Timer Clock Enable 0: Disable 1: Enable
14	USART1	R/W	USART1 Clock Enable 0: Disable 1: Enable
15	ADC3	R/W	ADC 3 Interface Clock Enable 0: Disable 1: Enable
31:16	Reserved		

5.5.8 APB1 peripheral clock enable register (RCM_APB1CLKEN)

Offset address: 0x1C

Reset value: 0x0000 0000

Access: Access in the form of word, half word and byte

Usually there is no wait cycle. However, when the peripheral on the APB1 bus is accessed, the waiting state will be inserted until the APB1 peripheral access ends.

All bits can be reset or cleared by software.

Note: When the peripheral clock is not enabled, the software cannot read the value of the peripheral register, and the value returned is always 0x0.

Field	Name	R/W	Description
0	TMR2	R/W	Timer 2 Clock Enable 0: Disable 1: Enable
1	TMR3	R/W	Timer 3 Clock Enable 0: Disable 1: Enable
2	TMR4	R/W	Timer 4 Clock Enable 0: Disable 1: Enable

Field	Name	R/W	Description
3	TMR5	R/W	Timer 5 Clock Enable 0: Disable 1: Enable
4	TMR6	R/W	Timer 6 Clock Enable 0: Disable 1: Enable
5	TMR7	R/W	Timer 7 Clock Enable 0: Disable 1: Enable
10:6	Reserved		
11	WWDT	R/W	Window Watchdog Clock Enable 0: Disable 1: Enable
13:12	Reserved		
14	SPI2	R/W	SPI 2 Clock Enable 0: Disable 1: Enable
15	SPI3	R/W	SPI 3 Clock Enable 0: Disable 1: Enable
16	Reserved		
17	USART2	R/W	USART 2 Clock Enable 0: Disable 1: Enable
18	USART3	R/W	USART 3 Clock Enable 0: Disable 1: Enable
19	UART4	R/W	UART 4 Clock Enable 0: Disable 1: Enable
20	UART5	R/W	UART 5 Clock Enable 0: Disable 1: Enable
21	I2C1	R/W	I2C1/3 Clock Enable 0: Disable 1: Enable
22	I2C2	R/W	I2C2/4 Clock Enable 0: Disable 1: Enable
23	USBD	R/W	USBD1/2 Clock Enable 0: Disable 1: Enable
24	Reserved		

Field	Name	R/W	Description
25	CAN1	R/W	CAN1 Clock Enable 0: Disable 1: Enable
26	CAN2	R/W	CAN2 Clock Enable 0: Disable 1: Enable
27	BAKP	R/W	Backup Interface Clock Enable 0: Disable 1: Enable
28	PMU	R/W	Power Interface Clock Enable 0: Disable 1: Enable
29	DAC	R/W	DAC Interface Clock Enable 0: Disable 1: Enable
31:30	Reserved		

5.5.9 Backup domain control register (RCM_BDCTRL)

Offset address: 0x20

Reset value: 0x0000 0000, which can be reset effectively only by backup domain

Access: Access in the form of word, half word and byte, with 0 to 3 wait cycles

When the register is accessed continuously, the waiting state will be inserted.

Note: Only when BPWEN bit in PMU_CTRL is set to 1, can LSEEN, LSEBCFG, RTCSRCSEL and RTCCLKEN be changed.

Field	Name	R/W	Description
0	LSEEN	R/W	Low-Speed External Clock Enable 0: Disable 1: Enable
1	LSERDYFLG	R	Low-Speed External Oscillator Oscillator Ready Flag When LSECLK is stable, this bit is set to 1 by hardware, and when it is unstable, it is cleared by hardware. 0: Not ready 1. Ready
2	LSEBCFG	R/W	Low-Speed External Clock Bypass Mode Configure Bypass mode refers to the mode in which external clock is used as the LSECLK clock source; otherwise the resonator is used as the LSECLK clock source. 0: Non-bypass mode 1: Bypass mode
7:3	Reserved		
9:8	RTCSRCSEL	R/W	RTC Clock Source Select First set the BDRST bit to reset the backup domain, and then select the RTC clock source. It is impossible to directly configure the register to modify.

Field	Name	R/W	Description
			00: No clock 01: LSECLK oscillator is used as RTC clock 10: LSICLK oscillator is used as RTC clock 11: HSECLK oscillator is used as RTC clock after 128 divided frequency
14:10	Reserved		
15	RTCCLKEN	R/W	RTC Clock Enable 0: Disable 1: Enable
16	BDRST	R/W	Backup Domain Software Reset Set 1 or clear 0 by software 0: Reset is not activated 1: Reset the whole backup domain
31:17	Reserved		

5.5.10 Control/State register (RCM_CSTS)

Offset address: 0x24

Reset value: 0x0C00 0000, except reset flag, all are cleared by system reset, and reset flag can only be cleared by power reset.

Access: Access in the form of word, half word and byte, with 0 to 3 wait cycles. When the register is accessed continuously, the waiting state will be inserted.

Field	Name	R/W	Description
0	LSIEN	R/W	Low-Speed Internal Oscillator Enable Set 1 or clear 0 by software. 0: Disable 1: Enable
1	LSIRDYFLG	R	Low-Speed Internal Oscillator Ready Flag When LSICLK is stable, this bit is set to 1 by hardware, and when it is unstable, it is cleared by hardware. 0: Not ready 1. Ready
23:2	Reserved		
24	RSTFLGCLR	R/W	Reset Flag Clear The reset flag is cleared by setting to 1 by software, including RSTFLGCLR. 0: No effect 1: Clear the reset flag
25	Reserved		
26	NRSTFLG	R/W	NRST PIN Reset Occur Flag When NRST pin is reset, it is set to 1 by hardware and cleared by software by writing RSTFLGCLR bit. 0: No NRST pin reset 1: NRST pin reset occurred

Field	Name	R/W	Description
27	PODRSTFLG	R/W	POR/PDR Reset Occur Flag This bit can be set to 1 by hardware; and cleared by software by writing RSTFLGCLR bit. 0: No power-on/power-down reset occurs 1: Power-on/power-down reset occurs
28	SWRSTFLG	R/W	Software Reset Occur Flag This bit can be set to 1 by hardware; and cleared by software by writing RSTFLGCLR bit. 0: No occurrence 1: Occurred
29	IWDTRSTFLG	R/W	Independent Watchdog Reset Occur Flag When independent watchdog reset occurs in V_{DD} area, it is set to 1 by hardware and cleared by software by writing RSTFLGCLR bit. 0: No occurrence 1: Occurred
30	WWDTRSTFLG	R/W	Window Watchdog Reset Occur Flag When window watchdog is reset, it is set to 1 by hardware and cleared by software by writing RSTFLGCLR bit. 0: No occurrence 1: Occurred
31	LPWRRSTFLG	R/W	Low Power Reset Occur Flag When low-power management is reset, it is set to 1 by hardware and cleared by software by writing RSTFLGCLR bit. 0: No occurrence 1: Occurred

6 Power Management Unit (PMU)

6.1 Full Name and Abbreviation Description of Terms

Table 32 Full Name and Abbreviation Description of Terms

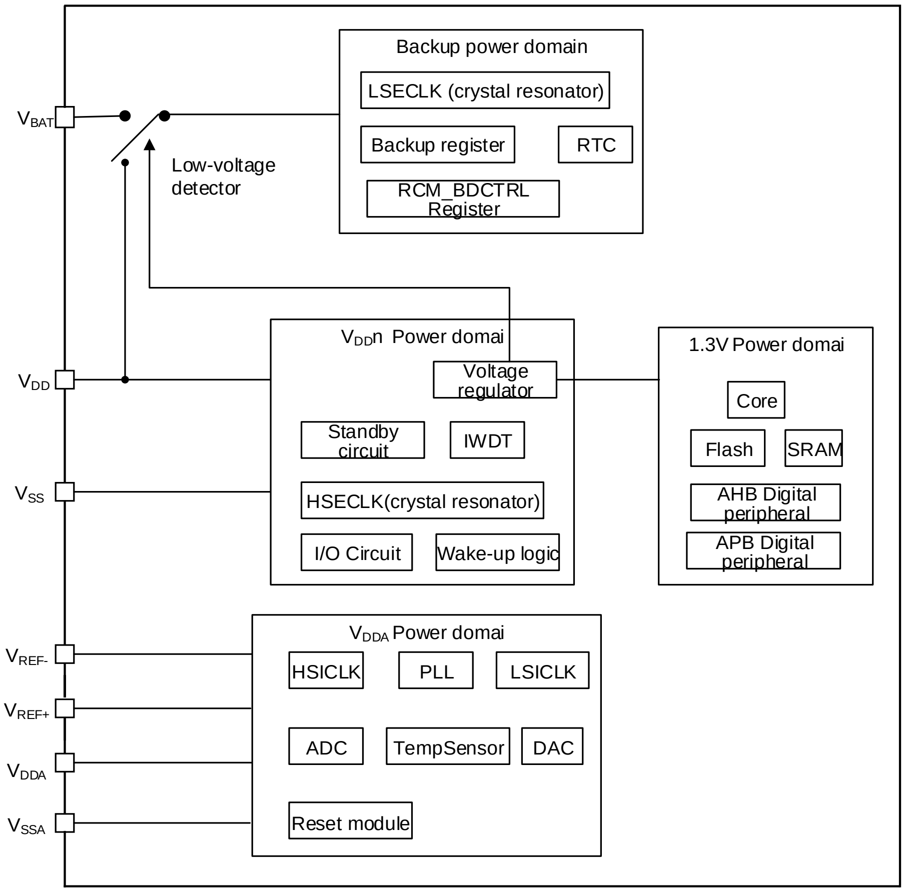
Full name in English	English abbreviation
Power Management Unit	PMU
Power On Reset	POR
Power Down Reset	PDR
Power Voltage Detector	PVD

6.2 Introduction

The power supply is the basis for stable operation of a system. The working voltage is 2.0~3.6V. It can provide 1.3V power supply through the built-in voltage regulator. If the main power V_{DD} is powered down, it can supply power to the backup power supply area through V_{BAT}.

6.3 Structure Block Diagram

Figure 8 Power Supply Control Structure Block Diagram



6.4 Functional Description

6.4.1 Power Domain

The power domain of the product includes: V_{DD} power domain, V_{DDA} power domain, 1.3V power domain and backup power domain.

6.4.1.1 V_{DD} power domain

Power supply is provided through V_{DD}/V_{SS} pins to power the voltage regulator, standby circuit, IWDT, HSECLK, I/O (except PC13, PC14, PC15 pins) and wake-up logic.

Voltage regulator

Power can be supplied to 1.3V power domain in the following operating modes:

- Normal mode: In this mode, 1.3V power supply area runs at full power
- Stop mode: In this mode, 1.3V power supply area works in low power state, all clocks are off, and peripherals stop work

- Standby mode: In this mode, the 1.3V power supply area stops power supply, and except for the standby circuit, the content of register and SRAM will be lost

6.4.1.2 V_{DDA} power domain

Power the ADC, DAC, HSICLK, LSICLK, TempSensor, PLL and reset module through VDDA/VSSA and VREF+/VREF- pins.

Independent ADC power supply and reference voltage

Independent ADC power supply can improve conversion accuracy, and the specific power pins are as follows:

- V_{DDA}: Power pin of ADC
- V_{SSA}: Independent power ground pin
- V_{REF+}/V_{REF-}: ADC reference voltage pin

6.4.1.3 1.3V power domain

The core, Flash, SRAM and digital peripherals are powered by voltage regulator.

6.4.1.4 Backup power domain

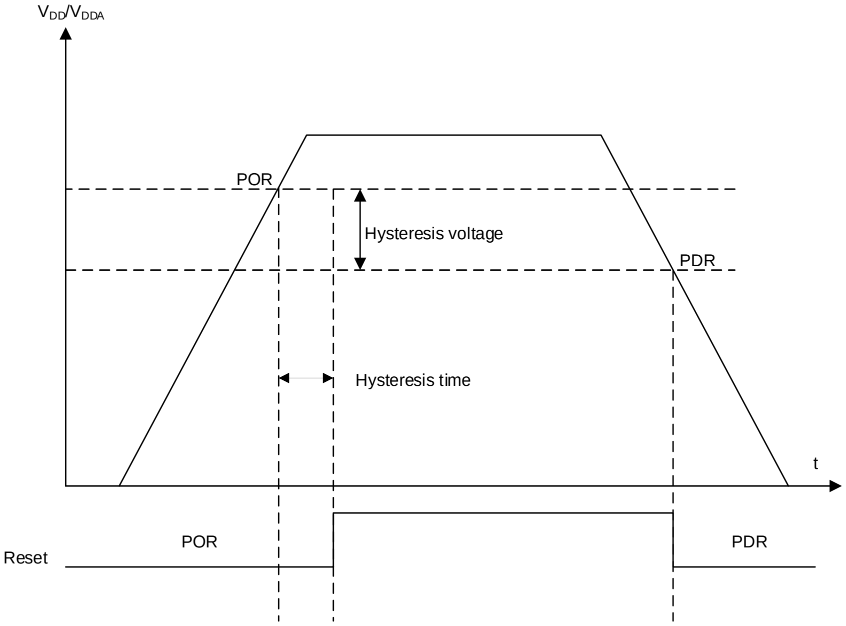
When V_{DD} exists, the backup power supply area is powered by V_{DD}. When V_{DD} is powered down, the backup power supply area is powered by V_{BAT}, which is used to save the content of backup register and maintain RTC function. Power the LSECLK crystal oscillator, RTC, backup register and RCM_BDCTRL register, PC13, PC14 and PC15.

6.4.2 Power Management

6.4.2.1 Power-on/power-down reset (POR and PDR)

When the V_{DD}/V_{DDA} is detected to be lower than the threshold voltage V_{POR} and V_{PDR}, the chip will automatically maintain the reset state. The waveform diagrams of power-on reset and power-down reset are as follows. For POR, PDR, hysteresis voltage and hysteresis time, please refer to the "Datasheet".

Figure 9 Power-on Reset and Power-down Reset Oscillogram



6.4.2.2 Power voltage detector (PVD)

A threshold can be set for PVD to monitor whether V_{DD}/V_{DDA} is higher or lower than the threshold. If interrupt is enabled, the interrupt can be triggered to process VDD/VDDA exceeding the threshold in advance. The usage of PVD is as follows:

（1） Set the PVDEN bit of the configuration register PMU_CTRL to 1 to enable PVD

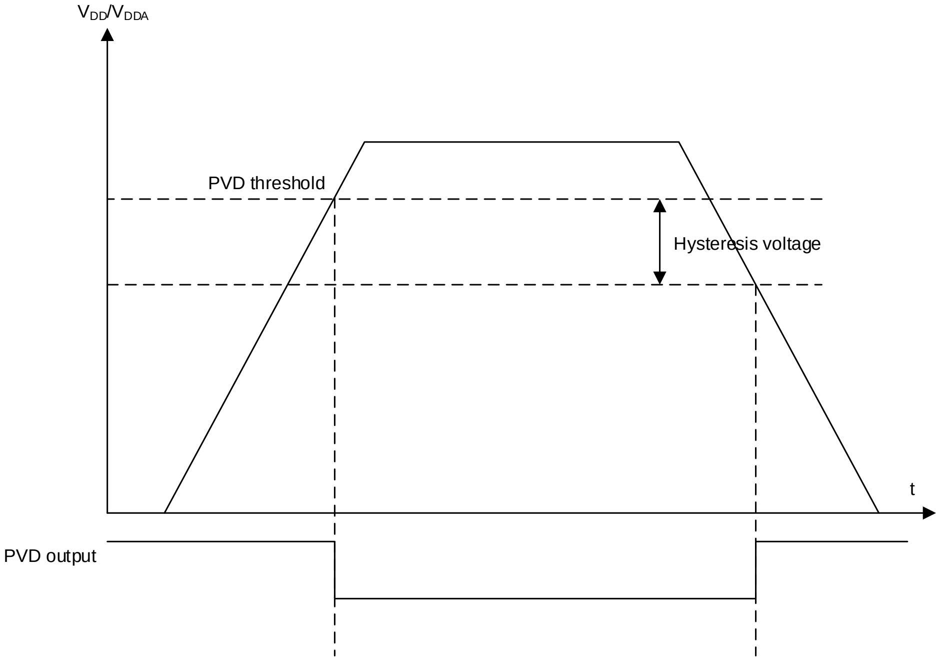
（2） Select the voltage threshold of PVD for the PLSEL[2:0] bit of the configuration register PMU_CTRL

（3） The PVDOFLG bit of the configuration register PMU_CSTS indicates the value of V_{DD} is higher or lower than the threshold of PVD

（4） When it is detected that V_{DD}/V_{DDA} is lower or higher than the threshold of PVD, PVD interrupt will be generated

The threshold waveform of PVD is shown below. Please see "Datasheet" for PVD threshold and hysteresis voltage.

Figure10 PVD Threshold Oscillogram



6.4.3 Power Consumption Control

6.4.3.1 Reduce the power consumption in low-power mode

There are three low-power modes: sleep mode, stop mode and standby mode. The power consumption is reduced by closing the core and clock source and setting the voltage regulator.

The power consumption, wake-up start time, wake-up mode and data storage of each low-power mode are different; the lower the power consumption is, the longer the wake-up time is, the less the wake-up mode is, the less the data saved are after wake-up; users can choose the most appropriate low-power mode according to their needs. The following table shows the difference among three low-power modes.

Table 32 Difference among "Sleep Mode, Stop Mode and Standby Mode"

Mode	Instruction	Entry mode	Wake-up mode	Voltage regulator	Effect on 1.3V area clock	Effect on V_{DD} area clock
Sleep	Arm® Cortex®-M3 core stops, and all peripherals including the core peripheral are still working	Call WFI instruction	Any interrupt	Open	Ony the core clock is turned off and it has no effect on other clocks and ADC clocks	None
		Call WFE instruction	Wake-up event	Open		None

Mode	Instruction	Entry mode	Wake-up mode	Voltage regulator	Effect on 1.3V area clock	Effect on V_{DD} area clock
Stop	All clocks have stopped	PDDSCFG and LPDSCFG bits +SLEEPDEEP bit +WFI or WFE	Any external interrupt	Turn on or be in low-power mode	Close clocks of all 1.3V areas	The oscillator of HSICLK and HSECLK is turned off
Standby	1.3V power off	PDDSCFG bit +SLEEPDEEP bit +WFI or WFE	Rising edge of WKUP pin, RTC alarm event, external reset on NRST pin, IWDT reset	Off		

Sleep mode

The characteristics of sleep mode are shown in the table below

Table 33 Characteristics of Sleep Mode

Characteristics	Instruction
Enter	Enter the sleep mode immediately by executing WFI or WFE instructions; When SLEEPONEINT is set to 0 and WFI or WFE instruction is executed, the system will enter the sleep mode immediately; when SLEEPONEINT is set to 1, the system will exit the interrupt program and then enter the sleep mode immediately.
Wake-up	If WFI instruction is executed to enter the sleep mode, wake up by any interrupt; If WFE instruction is executed to enter the sleep mode, wake up through an event.
Sleep	The core stops working, all peripherals are still running, and the data in the core registers and memory before sleep are saved.
Wake-up delay	None
After wake-up	If the system is woken up by interrupt, it will first enter the interrupt, then exit the interrupt, and then execute the program after WFI instruction. If the system is woken up by event, it will directly execute the program after WFE instruction.

Stop mode

The characteristics of stop mode are shown in the table below:

Table 34 Characteristics of Stop Mode

Characteristics	Instruction
Enter	SLEEPDEEP bit of the core register is set to 1, PDDSCFG bit of the register PMU_CTRL is set to 0, and when executing WFI or WFE instruction, the system will enter the stop mode immediately; When LPDSCFG bit of the register PMU_CTRL is set to 0, the voltage regulator is working in normal mode; when LPDSCFG bit of the register PMU_CTRL is set to 1, the voltage regulator is working in low-power mode.

Characteristics	Instruction
Wake-up	If WFI instruction is executed to enter the sleep mode, wake up by any interrupt; If WFE instruction is executed to enter the sleep mode, wake up through an event.
Stop	The core will stop working, the peripheral will stop working, and the data in the core register and memory before stop will be saved.
Wake-up delay	HSICLK oscillator wake-up time + voltage regulator wake-up time from low-power mode.
After wake-up	If the system is woken up by interrupt, it will first enter the interrupt, then exit the interrupt, and then execute the program after WFI instruction. If the system is woken up by event, it will directly execute the program after WFE instruction.

Standby mode

The characteristics of standby mode are shown in the table below:

Table 35 Standby Mode

Characteristics	Instruction
Enter	SLEEPDEEP bit of the core register is set to 1, PDDSCFG bit of the register PMU_CTRL is set to 1, WUEFLG bit is set to 0 and when executing WFI or WFE instruction, the system will enter the standby mode immediately.
Wake-up	Wake up by rising edge of WKUP pin, RTC alarm, wake-up, tamper event or NRST pin external reset and IWDT reset.
Standby	The core will stop working, the peripheral will stop working, and the data in the core register and memory will be lost.
Wake-up delay	Chip reset time.
After wake-up	The program starts executing from the beginning.

6.4.3.2 Reduce the power consumption in run mode

In the run mode, the power consumption can be reduced by reducing the system clock, closing or reducing the peripheral clock on the APB/AHB bus.

6.5 Register Address Mapping

Table 36 PMU Register Address Mapping Table

Register name	Description	Offset address
PMU_CTRL	Power control register	0x00
PMU_CSTS	Power control/state register	0x04

6.6 Register Functional Description

6.6.1 Power control register (PMU_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000 (cleared when waking up from standby mode)

Field	Name	R/W	Description
0	LPDSCFG	R/W	Low Power Deepsleep Configure Configure the working state of the voltage regulator in stop mode. 0: Enable 1: Low-power mode
1	PDDSCFG	R/W	Power Down Deep Sleep Configure When the CPU enters deep sleep, configure the voltage regulator state in standby and stop modes. 0: The voltage regulator is controlled by LPDSCFG bit when entering the stop mode 1: Enter standby mode
2	WUFLGCLR	RC_W1	Wakeup Flag Clear 0: Invalid 1: Clear the wake-up flag after 2 system clock cycles by writing 1
3	SBFLGCLR	RC_W1	Standby Flag Clear 0: Invalid 1: Write 1 to clear the standby flag
4	PVDEN	R/W	Power Voltage Detector Enable 0: Disable 1: Enable
7:5	PLSEL	R/W	PVD Level Select 0x0：2.2V 0x1：2.3V 0x2：2.4V 0x3：2.5V 0x4：2.6V 0x5：2.7V 0x6：2.8V 0x7：2.9V Note: See "Datasheet" for detailed instructions
8	BPWEN	R/W	Backup Domain Write Access Enable Backup area refers to RTC and backup register; write access is disabled after reset, and is allowed after writing 1. 0: Write is disabe 1: Write is enable
31:9	Reserved		

6.6.2 Power control/state register (PMU_CSTS)

Offset address: 0x04

Reset value: 0x0000 0000 (not cleared when waking up from standby mode)

Compared with the standard APB read, it requires extra APB cycle to read this register

Field	Name	R/W	Description
0	WUEFLG	R	Wakeup Event Flag This bit is set by hardware, indicating whether wake-up event or RTC alarm wake-up event occurs on WKUP pin 0: Not occur

Field	Name	R/W	Description
			1: Occurred Note: Enable the WKUP pin, and an event will be detected when the WKUP pin is at high level.
1	SBFLG	R	Standby Flag This bit is set to 1 by hardware, and can only be cleared by POR/PDR (power-on/power-down reset) or by setting the SBFLGCLR bit of the power supply control register (PMU_CTRL). 0: Not enter the standby mode 1: Have entered the standby mode
2	PVDOFLG	R	PVD Output Flag Indicate whether VDD/VDDA is higher than the PVD threshold selected by PLSEL[2:0] This bit is valid only when PVD is enabled by PVDEN BIT. 0: V_{DD}/V_{DDA} higher than PVD threshold 1: V_{DD}/V_{DDA} lower than PVD threshold Note: This bit is 0 after reset or when entering the standby mode (PVD stops work).
7:3	Reserved		
8	WKUPCFG	R/W	WKUP Pin Configure When WKUP is used as a normal I/O, the event on WKUP pin cannot wake up the CPU in standby mode; it can wake up CPU only when it is not used as a normal I/O. 0: Configure normal I/O 1: Can wake MCU Note: Clear this bit in system reset
31:9	Reserved		

7 Backup Register (BAKPR)

7.1 Introduction

The backup register can be used to store 84 bytes of data, including 42 16-bit registers. When V_{DD} is closed, the backup domain will be maintained power-on by V_{BAT}.

Wake up the system in standby mode. If the system is reset or the power supply is reset, the backup register will not be reset. BAKPR control register manages tamper detection and RTC check.

After BAKP is reset, access to the backup register and RTC will be disabled, and the backup domain (BAKPR) will be protected from possible accidental write access. If you want to re-enable the access to the backup register and RTC, operate according to the following steps:

- Enable the power supply and standby interface clock by setting PMU and BAKP bits in RCM_APB1CLKEN register
- Enable the access to the backup register and RTC by setting BPWEN bit in PMU_CTRL power control register

7.2 Main Characteristics

（1） 84-byte data register

（2） The state/control register is used to manage the tamper detection pull-up input with interrupt function

（3） Check register, which can store RTC calibration value

（4） Output the RTC calibration clock, RTC alarm pulse or second pulse on tamper pin PC13 (TAMPER) (when the pin is not used for tamper detection)

7.3 Functional Description

7.3.1 Intrusion Detection

Judge whether tamper event is generated according to whether the signal on the TAMPER pin changes. Intrusion detection event can reset all data backup registers. In order to avoid the loss of tamper events, detect the signal and also detect the edge detection signal and tamper detection enable bit so that the tamper events before detection can be detected. When the TPALCFG bit is set, if the tamper pin is already at a effective level before enabling, an additional tamper event will be generated after the tamper pin is enabled. If TPIEN bit of BAKPR_CSTS register is also set, an interrupt will be generated when an tamper detection event occurs.

Disable the tamper pin after an tamper event is detected and cleared. If you want to re-enable the tamper detection function, to avoid that there is still tamper detection event on tamper pin when the software writes backup data BAKPR_DATAx register, it is required to set TPFCFG bit of BAKPR_CTRL register (equivalent to tamper pin detection) before writing the backup data BAKPR_DATAx register.

Note: The tamper detection is still active when V_{DD} is powered off. The tamper pin should be externally connected to the correct level to prevent the reset data backup register from being reset.

7.3.2 RTC Calibration

Enable RTC calibration by configuring the CALCOEN bit of RTC clock calibration BAKPR_CLKCAL register.

RTC clock can be output to the tamper pin through 64 divided frequency.

7.4 Register Address Mapping

Table 37 BAKPR Register Address Mapping

Register name	Description	Offset address
BAKPR_DATAx(x=1..10)	Backup data register 1	0x04+4(x-1)
BAKPR_CLKCAL	RTC clock calibration register	0x2C
BAKPR_CTRL	Backup control register	0x30
BAKPR_CSTS	Backup control/state register	0x34
BAKPR_DATAx(x=11…42)	Backup data register 11	0x40+4(x-1)

7.5 Register Functional Description

Peripheral registers can be accessed by half word (16 bits) or word (32 bits).

7.5.1 Backup data register x (BAKPR_DATAx) (x=1…10, 11…42)

Offset address: From 0x04 to 0x28, from 0x40 to 0xBC

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	DATA	R/W	User Backup Data In the presence of V_{BAT} power supply, BAKPR_DATAx register cannot be reset through power reset, system reset and standby mode wake-up reset, and can only be reset by resetting the backup domain or tamper event.

7.5.2 RTC clock calibration register (BAKPR_CLKCAL)

Offset address: 0x2C

Reset value: 0x0000 0000

Field	Name	R/W	Description
6:0	CALVALUE	R/W	Calibration Value Setup Reduce RTC clock by skipping the clock pulse count of RTC, to realize calibration. This value indicates the pulse count of multiple clocks that will be ignored every 2^{20} clock pulses, which can be slowed down from 0 to 121ppm.
7	CALCOEN	R/W	Calibration Clock Output Enable 0: No output 1: For the RTC clock after the tamper pin outputs 64 divided frequency, if LSECLK is 32.768KHz, the output signal frequency is 512Hz. When the CALCOEN bit is set, the tamper detection function needs to be disabled to avoid unnecessary tamper signal detected. Note: This bit will be cleared when V_{DD} is powered off.
8	ASPOEN	R/W	Alarm or Second Pulse Output Enable 0: Disable 1: Output RTC entry alarm or second pulse signal on tamper pin The duration of output pulse is 1 RTC clock cycle; when setting the ASPOEN bit, the tamper detection function should be disabled. Note: This bit can be clered only by backup domain reset.
9	ASPOSEL	R/W	Alarm or Second Pulse Output Select This bit can select the tamper pin to output RTC second pulse signal or alarm pulse signal 0: Output RTC alarm pulse 1: Output RTC second pulse Note: This bit can be clered only by backup domain reset.
31:10	Reserved		

7.5.3 Backup control register (BAKPR_CTRL)

Offset address: 0x30

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	TPFCFG	R/W	TAMPER Pin Function Configure 0: Tamper pin is used as general-purpose IO port 1: Tamper pin is shared in tamper detection
1	TPALCFG	R/W	TAMPER Pin Active Level Configure Select the effective level detected by the tamper pin to reset all the data backup registers. 0: High level 1: Low level
31:2	Reserved		

Note: Setting TPALCFG and TPFCFG bits at the same time is always secure. However, a false tamper event will be generated if both are cleared at the same time. Therefore, it is recommended to change the state of TPALCFG bit only when TPFCFG is 0.

7.5.4 Backup control/state register (BAKPR_CSTS)

Offset address: 0x34

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	TECLR	W	Tamper Event Flag Clear This bit is write-only, and the read-out value is 0 0: Invalid 1: Clear the tamper detection event flag and reset the tamper detection function
1	TICLR	W	Tamper Interrupt Flag Clear This bit is write-only, and the read-out value is 0 0: Invalid 1: Clear the tamper detection interrupt and interrupt flag
2	TPIEN	R/W	TAMPER Pin Interrupt Enable This bit is reset only after system reset or wake-up from standby mode. Tamper interrupt cannot wake up the system core in low-power mode. 0: Disable 1: Enable (TPFCFG bit must be set)
7:3	Reserved		
8	TEFLG	R	TAMPER Event Occur Flag This bit is set by hardware when an tamper event is detected and it can be cleared by writing 1 to TECLR bit 0: No tamper event 1: Tamper event detected Note: The tamper event can reset all backup data registers. If the bit is 1, all backup data registers will remain reset, and the backup data cannot be written successfully.
9	TIFLG	R	TAMPER Interrupt Occur Flag When the TPIEN bit is set and an tamper event is detected, this bit is set by hardware and cleared by writing 1 to the TICLR bit; this bit is reset only after the system is reset or woken up from standby mode. 0: No tamper interrupt 1: Tamper interrupt occurred
31:10	Reserved		

8 Nested Vector Interrupt Controller (NVIC)

8.1 Full Name and Abbreviation Description of Terms

Table 38 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Non Maskable Interrupt	NMI

8.2 Introduction

The Cortex-M3 core in the product integrates nested vectored interrupt controller (NVIC), which is closely coupled with the core, and can handle exceptions and interrupts and power management control efficiently and with low delay. Please see *Cortex-M3 Technical Reference Manual* for more instructions about NVIC.

8.3 Main Characteristics

（1） 65 maskable interrupt channels (excluding 16 Arm® Cortex®-M3 interrupt lines)

（2） 16 programmable priority levels (use 4-bit interrupt priority level)

（3） Low-delay exception and interrupt processing

（4） Power management control

（5） Realization of system control register

8.4 Interrupt and Exception Vector Table

Table 39 APM32E103xCxE Interrupt and Exception Vector Table

Exception type	Vector No.	Priority	Vector address	Description
-	-	-	0x0000_0000	Reserved
Reset	-	-3	0x0000_0004	Reset
NMI	-	-2	0x0000_0008	Non-maskable interrupt
HardFault	-	-1	0x0000_000C	Various hardware faults
MemManage	-	Can be set	0x0000_0010	Memory management
BusFault	-	Can be set	0x0000_0014	-
UsageFault	-	Can be set	0x0000_0018	-
-	-	-	0x0000_001C- 0x0000_002B	Reserved

Exception type	Vector No.	Priority	Vector address	Description
SVCall	-	Can be set	0x0000_002C	SWI instruction realizes system service revoking
Debug Monitor	-	Can be set	0x0000_0030	Debug monitor
-	-	-	0x0000_0034	Reserved
PendSV	-	Can be set	0x0000_0038	Pending system service request
SysTick	-	Can be set	0x0000_003C	System tick timer
WWDT	0	Can be set	0x0000_0040	Window watchdog interrupt
PVD	1	Can be set	0x0000_0044	Power voltage detection interrupt
TAMPER	2	Can be set	0x0000_0048	Tamper detection interrupt
RTC	3	Can be set	0x0000_004C	RTC interrupt
FLASH	4	Can be set	0x0000_0050	Flash memory global interrupt
RCM	5	Can be set	0x0000_0054	RCM interrupt
EINT0	6	Can be set	0x0000_0058	EINT Line 0 interrupt
EINT1	7	Can be set	0x0000_005C	EINT Line 1 interrupt
EINT2	8	Can be set	0x0000_0060	EINT Line 2 interrupt
EINT3	9	Can be set	0x0000_0064	EINT Line 3 interrupt
EINT4	10	Can be set	0x0000_0068	EINT Line 4 interrupt
DMA1_CH1	11	Can be set	0x0000_006C	DMA1 channel 1 global interrupt
DMA1_CH2	12	Can be set	0x0000_0070	DMA1 channel 2 global interrupt
DMA1_CH3	13	Can be set	0x0000_0074	DMA1 channel 3 global interrupt
DMA1_CH4	14	Can be set	0x0000_0078	DMA1 channel 4 global interrupt
DMA1_CH5	15	Can be set	0x0000_007C	DMA1 channel 5 global interrupt
DMA1_CH6	16	Can be set	0x0000_0080	DMA1 channel 6 global interrupt
DMA1_CH7	17	Can be set	0x0000_0084	DMA1 channel 7 global interrupt
ADC1/2	18	Can be set	0x0000_0088	ADC1 and ADC2 global interrupt

Exception type	Vector No.	Priority	Vector address	Description
USBD1_HP_CAN1_TX	19	Can be set	0x0000_008C	USBD1 high-priority interrupt/CAN1 sending interrupt
USBD1_LP_CAN1_RX0	20	Can be set	0x0000_0090	USBD1 low-priority interrupt/CAN1 receiving 0 interrupt
CAN1_RX1	21	Can be set	0x0000_0094	CAN1 receiving 1 interrupt
CAN1_SCE	22	Can be set	0x0000_0098	CAN1 SCE interrupt
EINT9_5	23	Can be set	0x0000_009C	EINT line [9:5] interrupt
TMR1_BRK	24	Can be set	0x0000_00A0	TMR1 braking interrupt
TMR1_UP	25	Can be set	0x0000_00A4	TMR1 update interrupt
TMR1_TRG_COM	26	Can be set	0x0000_00A8	TMR1 trigger and communication interrupt
TMR1_CC	27	Can be set	0x0000_00AC	TMR1 capture/compare interrupt
TMR2	28	Can be set	0x0000_00B0	TMR2 interrupt
TMR3	29	Can be set	0x0000_00B4	TMR3 interrupt
TMR4	30	Can be set	0x0000_00B8	TMR4 interrupt
I2C1_EV	31	Can be set	0x0000_00BC	I2C1 event interrupt
I2C1_ER	32	Can be set	0x0000_00C0	I2C1 error interrupt
I2C2_EV	33	Can be set	0x0000_00C4	I2C2 event interrupt
I2C2_ER	34	Can be set	0x0000_00C8	I2C2 error interrupt
SPI1	35	Can be set	0x0000_00CC	SPI1 interrupt
SPI2	36	Can be set	0x0000_00D0	SPI2 interrupt
USART1	37	Can be set	0x0000_00D4	USART1 interrupt
USART2	38	Can be set	0x0000_00D8	USART2 interrupt
USART3	39	Can be set	0x0000_00DC	USART3 interrupt
EINT15_10	40	Can be set	0x0000_00E0	EINT line [15:10] interrupt
RTC_Alarm	41	Can be set	0x0000_00E4	RTC alarm interrupt
USBD_WakeUp	42	Can be set	0x0000_00E8	USBD_OTG wake-up interrupt
TMR8_BRK	43	Can be set	0x0000_00EC	TMR8 braking interrupt
TMR8_UP	44	Can be set	0x0000_00F0	TMR8 update interruupt
TMR8_TRG_COM	45	Can be set	0x0000_00F4	TMR8 trigger and communication interrupt

Exception type	Vector No.	Priority	Vector address	Description
TMR8_CC	46	Can be set	0x0000_00F8	TMR8 capture/compare interrupt
ADC3	47	Can be set	0x0000_00FC	ADC3 global interrupt
EMMC	48	Can be set	0x0000_0100	EMMC interrupt
SDIO	49	Can be set	0x0000_0104	SDIO interrupt
TMR5	50	Can be set	0x0000_0108	TMR5 interrupt
SPI3	51	Can be set	0x0000_010C	SPI3 interrupt
UART4	52	Can be set	0x0000_0110	UART4 interrupt
UART5	53	Can be set	0x0000_0114	UART5 interrupt
TMR6	54	Can be set	0x0000_0118	TMR6 interrupt
TMR7	55	Can be set	0x0000_011C	TMR7 interrupt
DMA2_CH1	56	Can be set	0x0000_0120	DMA2 channel 1 interrupt
DMA2_CH2	57	Can be set	0x0000_0124	DMA2 channel 2 interrupt
DMA2_CH3	58	Can be set	0x0000_0128	DMA2 channel 3 interrupt
DMA2_CH4/5	59	Can be set	0x0000_012C	DMA2 channel 4/5 interrupt
-	-	-	0x0000_0130-0x0000_0133	Reserved
USBD2_HP_CAN2_TX	61	Can be set	0x0000_0134	USBD2 high-priority interrupt/CAN2 sending interrupt
USBD2_LP_CAN2_RX0	62	Can be set	0x0000_0138	USBD2 low-priority interrupt/CAN2 receiving 0 interrupt
CAN2_RX1	63	Can be set	0x0000_013C	CAN2 receiving 1 interrupt
CAN2_SCE	64	Can be set	0x0000_0140	CAN2 SCE interrupt

9 External Interrupt/Event Controller (EINT)

9.1 Introduction

The interrupts/events contain internal interrupt/event and external interrupt/event. In this manual, external interrupt refers to the interrupt/event caused by I/O pin input signal, which is EINTx in interrupt vector table; other interrupts are internal interrupts/events.

The events can be divided into hardware events and software events. Hardware events are generated by external/core hardware signals, while software events are generated by instructions.

Interrupts need to go through the interrupt handler function to realize the work to be processed, while events do not need to go through interrupt handler function, and the preset work can be triggered by hardware. The external events output pulse through events such as GPIO, while the internal events trigger another TMR to work, for example, through update event of one TMR.

9.2 Functional Description

9.2.1 "External Interrupt and Event" Classification and Difference Points

"External interrupt and event" can be classified into external hardware interrupt, external hardware event, external software event and external software interrupt according to trigger source, configuration and execution process. The difference points are shown in the table below:

Table 40 "External Interrupt and Event" Classification and Difference Points

Name	Trigger source	Configuration and execution process
External hardware interrupt	External signal	(1) Set the trigger mode, allow the interrupt request, and enable corresponding peripheral interrupt line (enable in NVIC); (2) When an edge consistent with the configuration is generated on the external interrupt line, an interrupt request will be generated, and the corresponding suspend bit will be set to 1. Write 1 to the corresponding bit of the pending register and the interrupt request will be cleared.
External hardware event	External signal	(1) Set the trigger mode and enable the event line; (2) When an edge consistent with the configuration is generated on the external interrupt line, one event request pulse will be generated, and the corresponding pending bit will not be set to 1.

Name	Trigger source	Configuration and execution process
External software request	Software interrupt register/transmission event (SEV) instruction	(1) Enable the event line; (2) Write 1 to the software interrupt event register of the corresponding event line to generate an event request pulse, and the corresponding pending bit will not be set to 1.
External software interrupt	Software interrupt register	(1) Allow interrupt request, and enable the corresponding peripheral interrupt line (enable in NVIC); (2) Write 1 to the software interrupt event register of the corresponding event line to generate an interrupt request, the corresponding pending bit will be set to 1; write 1 to the corresponding bit of the pending register and the interrupt request will be cleared.

9.2.2 Core Wake-up

Using WFI and WFE instructions can make the core stop working. When WFI instruction is used, any interrupt can wake up the core; when WFE instruction is used, the core can be wakened up by event.

When interrupt is used for wake-up, the interrupt handler function will be triggered, and normal interrupt configuration can wake up the core. When an event is used to wake up the core, the interrupt handler function will not be triggered, which will reduce the wake-up time, and the configuration method is:

（1） It can trigger an internal interrupt (internal hardware event) but cannot trigger the interrupt handler function for wake-up

- It can enable an internal interrupt in the peripheral, but cannot enable the corresponding interrupt in NVIC to avoid triggering the interrupt handler function
- Enable SEVONPEND bit in the system controller of the core, and execute WFE instruction to make the core enter sleep mode
- Generate an interrupt to wake up the core; when the core recovers from WFE, it is required to clear the pending bit of corresponding peripheral interrupt and the pending bit of peripheral NVIC interrupt channel (clear the pending register in the NVIC interrupt)

（2） Wake up through EINT line events (external hardware event)

- Configure EINT line as the event mode
- Execute WFE instruction to make the core enter the sleep mode
- Generate an interrupt to wake up the core; when the CPU recovers from WFE, since the pending bit of corresponding event line is not set, it is unnecessary to clear the interrupt pending bit of corresponding peripheral or the NVIC interrupt channel pending bit

9.2.2.1 Event wake-up

It can trigger an internal interrupt (internal hardware event) but cannot trigger the interrupt handler function for wake-up

（1） Enable an internal interrupt in the peripheral, but do not enable the corresponding interrupt in NVIC to avoid triggering the interrupt handler function;

（2） Enable SEVONPEND bit in the system controller of the core, and execute WFE instruction to make the core enter sleep mode;

（3） Generate an interrupt to wake up the core; when the core recovers from WFE, it is required to clear the pending bit of corresponding peripheral interrupt and the pending bit of peripheral NVIC interrupt channel (clear the pending register in the NVIC interrupt).

Wake up through EINT line events (external hardware event)

（1） Configure EINT line as the event mode;

（2） Execute WFE instruction to make the core enter the sleep mode;

（3） Generate an interrupt to wake up the core; when the CPU recovers from WFE, since the pending bit of corresponding event line is not set, it is unnecessary to clear the interrupt pending bit of corresponding peripheral or the NVIC interrupt channel pending bit.

9.2.3 External Interrupt and Event Line Mapping

Table 41 External Interrupt and Event Line Mapping

External Interrupt and Event Channel Name	External Interrupt and Event Line No.
PA0/PB0/PC0/PE0/PF0/PG0	EINT 0
PA1/PB1/PC1/PE1/PF1/PG0	EINT 1
…	…
PA15/PB15/PC15/PE15/PF15/PG0	EINT 15
PVD output	EINT 16
RTC Alarm event	EINT 17
USBD wake-up event	EINT 18
Reserved	EINT 19
Reserved	EINT 20
Reserved	EINT 21
Reserved	EINT 22
Reserved	EINT 23
Reserved	EINT 24
Reserved	EINT 25
Reserved	EINT 26
Reserved	EINT 27
Reserved	EINT 28

External Interrupt and Event Channel Name	External Interrupt and Event Line No.
Reserved	EINT 29
Reserved	EINT 30
Reserved	EINT 31

9.3 Register Address Mapping

Table 42 External Interrupt/Event Controller Register Mapping

Register name	Description	Offset address
EINT_IMASK	Interrupt mask register	0x00
EINT_EMASK	Event mask register	0x04
EINT_RTEN	Enable the rising edge trigger selection register	0x08
EINT_FTEN	Enable the falling edge trigger selection register	0x0C
EINT_SWINTE	Software interrupt event register	0x10
EINT_IPEND	Interrupt pending register	0x14

9.4 Register Functional Description

9.4.1 Interrupt mask register (EINT_IMASK)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
18:0	IMASKx	R/W	Interrupt Request Mask on Line x 0: Mask 1: Open
31:19	Reserved		

9.4.2 Event mask register (EINT_EMASK)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
18:0	EMASKx	R/W	Event Request Mask on Line x 0: Mask 1: Open
31:19	Reserved		

9.4.3 Enable the rising edge trigger selection register (EINT_RTEN)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
18:0	RTENx	R/W	Rising Trigger Event and Interrupt Enable of Line x 0: Disable 1: Enable
31:19	Reserved		

Note: Since the external wake-up lines are edge triggered, there should be no burr signal on these lines; when writing EINT_RTEN register, if the rising edge signal is on the external interrupt line, it will not be recognized and the set pending bit will not be set; in the same interrupt line, the rising edge trigger and falling edge trigger can be set at the same time.

9.4.4 Enable the falling edge trigger selection register (EINT_FTEN)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
18:0	FTENx	R/W	Falling Trigger Event Enable of Line x 0: Disable (interrupt and event) 1: Enable (interrupt and event)
31:19	Reserved		

Note: Since the external wake-up lines are edge triggered, there should be no burr signal on these lines; when writing EINT_FTEN register, if the rising edge signal is on the external interrupt line, it will not be recognized and the set pending bit will not be set; in the same interrupt line, the rising edge trigger and falling edge trigger can be set at the same time.

9.4.5 Software interrupt event register (EINT_SWINTE)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
18:0	SWINTEx	R/W	Software Interrupt Event on Line x This bit can be set to 1 by software, and be cleared by writing 1 to the corresponding bit of EINT_IPEND. When this bit is 0, the pending bit of EINT_IPEND can be set by writing 1. If EINT_IMASK (EINT_EMASK) is set to open the interrupt (event) request, an interrupt (event) will be generated. 0: No effect 1: Software generates an interrupt (event)
31:19	Reserved		

9.4.6 Interrupt pending register (EINT_IPEND)

Offset address: 0x14

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
18:0	IPENDx	RC_W1	Interrupt Pending Occur of Line x Flag When a trigger request on the corresponding edge of EINT occurs on an external interrupt line, it will be set to 1 by hardware; it can be cleared by changing the polarity of the edge detection or by writing 1 to this bit.
31:19	Reserved		

10 Direct Memory Access (DMA)

10.1 Full Name and Abbreviation Description of Terms

Table 43 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Global	G
Transfer	T
Half	H
Complete	C
Error	E
Channel	CH
Circular	CIR
Peripheral	PER
Increment	I
Memory	M
Priority	PRI
Number	N
Address	ADDR

10.2 Introduction

DMA (Direct Memory Access) can realize high-speed data transmission between peripheral devices and memory or between memory and memory without CPU intervention, thus saving CPU resources for other operations.

The product has two DMA controllers, DMA1 has 7 channels and DMA2 has 5 channes. Each channel can manage multiple DMA requests, but each channel can only respond to one DMA request at the same time. Each channel can set priority, and the arbiter can coordinate the priority of corresponding DMA requests of each DMA channel according to the priority of the channels.

10.3 Main Characteristics

（1） DMA1 has 7 channels, and DMA2 has 5 channels

（2） There are three data transmission modes: peripheral to memory, memory to peripheral, memory to memory

（3） Each channel has a special hardware DMA request for connection

（4） Support software priority and hardware priority when multiple requests occur at the same time

（5） Each channel has three event flags and independent interrupts

（6） Support circular transmission mode

（7） The number of data transmission is programmable, up to 65535

10.4 Functional Description

10.4.1 DMA Request

If the peripheral or memory needs to use DMA to transmit data, it is required to first send DMA request and wait for DMA approval before data transmission.

DMA has 12 channels, DMA1 has 7 and DMA2 has 5. Each channel is connected with different peripherals, and each channel has three event flags (DMA half transmission, DMA transmission completion and DMA transmission error). The logic of the three event flags may become a separate interrupt request, and they all support software triggering.

When multiple peripherals request the same channel, it is required to configure the corresponding register to turn on or off the request of each peripheral, so as to ensure that only one peripheral request can be turned on in a channel.

Table 44 DMA1 Request Mapping Table

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
TMR1	—	TMR1_CH1	TMR1_CH2	TMR1_CH4 TMR1_TRIG TMR1_COM	TMR1_UP	TMR1_CH3	—
TMR2	TMR2_CH3	TMR2_UP	—	—	TMR2_CH1	—	TMR2_CH2 TMR2_CH4
TMR3	—	TMR3_CH3	TMR3_CH4 TMR3_UP	—	—	TMR3_CH1 TMR3_TRIG	—
TMR4	TMR4_CH1	—	—	TMR4_CH2	TMR4_CH3	—	TMR4_UP
ADC1	ADC1	—	—	—	—	—	—
SPI/I2S	—	SPI1_RX	SPI1_TX	SPI/I2S2_RX	SPI/I2S2_TX	—	—
USART	—	USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
I2C	—	—	—	I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX

Table 45 DMA2 Request Mapping Table

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
TMR5	TMR5_CH4 TMR5_TRIG	TMR5_CH3 TMR5_UP	—	TMR5_CH2	TMR5_CH1

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
TMR6/DAC channel 1	—	—	TMR6_UP/ DAC channel 1	—	—
TMR7/DAC channel 2	—	—	—	TMR7_UP/ DAC channel 2	—
TMR8	TMR8_CH3 TMR8_UP	TMR8_CH4 TMR8_TRGI TMR8_COM	TMR8_CH1	—	TMR8_CH2
ADC3	—	—	—	—	ADC3
SPI/I2S3	SPI/I2S3_RX	SPI/I2S3_TX	—	—	—
UART4	—	—	UART4_RX	—	UART4_TX
SDIO	—	—	—	SDIO	—

10.4.2 DMA Channel

10.4.2.1 Transmission data are programmable

The data transmitted by DMA are programmable, up to 65535, and the transmission data bit width of peripherals and memory can be set by configuring PERSIZE bit and MEMSIZE bit of DMA_CHCFGx register.

10.4.2.2 Transmission width and alignment method are programmable

Programmable data transmission width DMA transmission operations:

Figure 11 Transmission Width with Source of 8bits and Target of 8bits

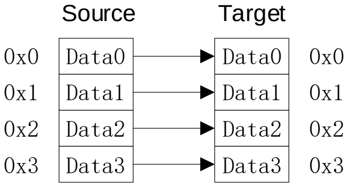


Figure 12 Transmission Width with Source of 8bits and Target of 16bits

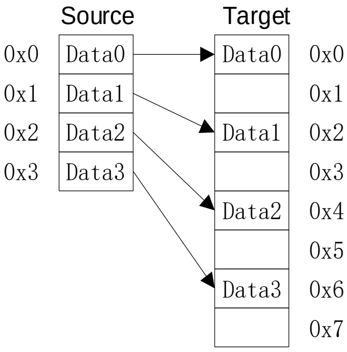


Figure 13 Transmission Width with Source of 8bits and Target of 32bits

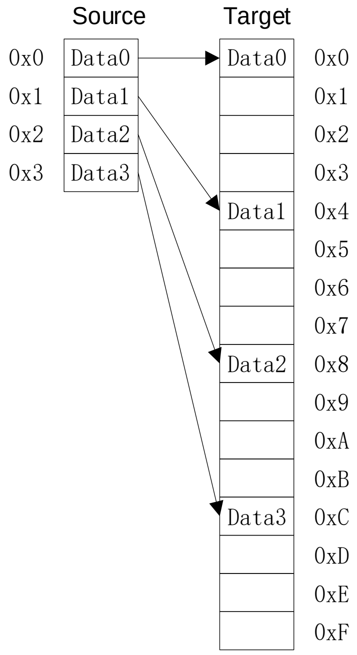


Figure 14Transmission Width with Source of 32bits and Target of 8bits

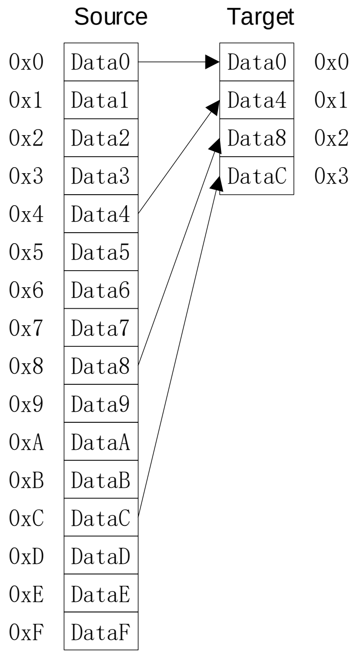


Figure 15Transmission Width with Source of 16bits and Target of 16bits

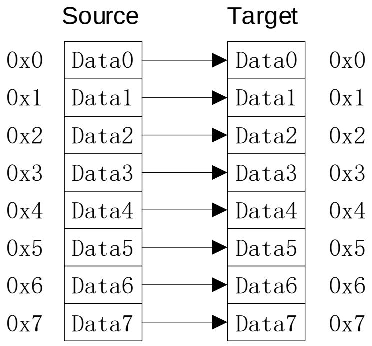


Figure 16 Transmission Width with Source of 16bits and Target of 32bits

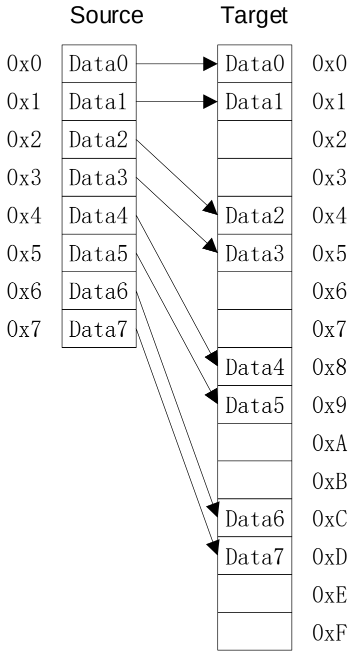
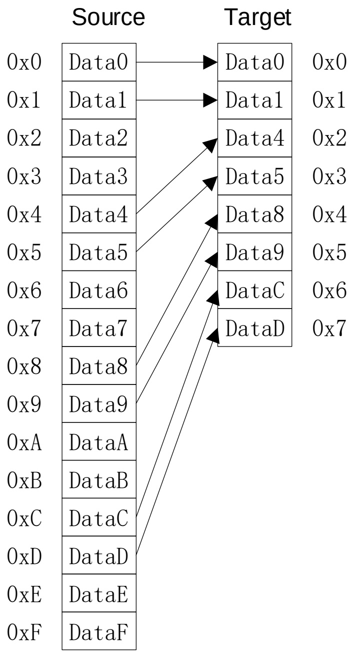


Figure 17 Transmission Width with Source of 32bits and Target of 16bits



10.4.2.3 Address setting

The transmission address supports two modes: fixed mode and pointer increment mode.

Transmission address pointer increment mode

The automatic pointer increment of peripheral and memory is completed through the PERIMODE bit and MIMODE bit of configuration register DMA_CHCFGx. The next address to be transmitted is the one by adding the increment to the previous address. The increment depends on the selected data width.

10.4.2.4 Transmission mode

There are two channel configuration modes: non-circular mode and circular mode.

Non-circular mode

When the data transmission is finished, the DMA operation will not be performed any more, and the new DMA transmission will be started. When the DMA channel is not working, the register DMA_CHNDATAx will rewrite the transmission value.

Circular mode

After data transmission, the content of the register DMA_CHNDATAx will be automatically reloaded to the previously configured value, and the peripheral address register DMA_CHPADDRx and the memory address register

DMA_CHMADDRx will also be reloaded as the initial base address.

The configuration method is as follows:

- Set the CIRMODE bit of the configuration register DMA_CHCFGx to 1 to turn on the circular mode;
- This mode is used to process continuous peripheral requests. When the number of data transmission becomes 0, it will automatically return to the initial value and continue DMA operation until the CIRMODE bit is cleared and the system exits the circular mode.

10.4.2.5 DMA request priority setting

Arbitrator

When multiple DMA channel requests occur, an arbiter is needed to manage the response sequence. Management is divided into two stages: the first stage is software stage, which is divided into the highest, high, medium and low priority; the second stage is hardware stage, and under the condition of the same software priority, the lower the channel number is, the higher the priority is.

10.4.2.6 Transmission direction

Support three directions: from memory to memory, from memory to peripheral, and from peripheral to memory.

If the write operation (target address) is performed on the memory, the memory includes internal SRAM, external RAM supported by EMMC (such as external SRAM, SDRAM) and NORFLASH; if the read operation (source address) is performed on the memory, the address includes internal FLASH, internal SRAM, RAM supported by EMMC, and NORFLASH.

Examples of "from memory to memory" configuration are as follows:

- The M2MMODE bit of the configuration register DMA_CHCFGx is set to put the memory to the memory mode;
- The DMA operation in this mode is performed under the condition of no peripheral request. The CHEN bit of the configuration register DMA_CHCFGx is set to 1, and after the channel is opened, the data transmission will start and when the transmission quantity register DMA_CHNDATAx becomes 0, the transmission is over.

10.4.3 Interrupt

Each DMA channel has three types of interrupt events, which are half transmission (HT), transmission completion (TC) and transmission error (TE).

（1） The interrupt event flag bit for half transmission is HTFLG, and the interrupt enable control bit is HTINTEN

（2） The interrupt event flag bit for transmission completion is TCFLG, and the interrupt enable control bit is TCINTEN

（3） The interrupt event flag bit for transmission error is TERRFLG, and the interrupt enable control bit is TERRINTEN

10.5 Register Address Mapping

Table 46 Register Address Mapping

Register name	Description	Offset address
DMA_INTSTS	DMA interrupt state register	0x00
DMA_INTFCLR	DMA interrupt flag clear register	0x04
DMA_CHCFGx	DMA Channel x configuration register	0x08+20 x
DMA_CHNDATAx	DMA Channel x transmission quantity register	0x0C+20 x
DMA_CHPADDRx	DMA Channel x peripheral address register	0x10+20 x
DMA_CHMADDRx	DMA Channel x memory address register	0x14+20 x

10.6 Register Functional Description

10.6.1 DMA interrupt state register (DMA_INTSTS)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
24,20,16, 12,8,4,0	GINTFLGx	R	Channel x Global Interrupt Occur Flag (x=1…7) Indicate whether TC, HT or TE interrupt is generated on the channel; these bits are set to 1 by hardware; write 1 and clear on the corresponding bit of DMA_INTFCLR. 0: Not generate 1: Generate
25,21,17, 13,9,5,1	TCFLGx	R	Channel x All Transfer Complete Flag (x=1..7) Indicate whether the transmission completion interrupt (TC) is generated on the channel; these bits are set to 1 by hardware; write 1 and clear on the corresponding bit of DMA_INTFCLR. 0: Not completed 1: Completed
26,22,18, 14,10,6,2	HTFLGx	R	Channel x Half Transfer Complete Flag (x=1..7) Indicate whether the half transmission interrupt (HT) is generated on the channel; these bits are set to 1 by hardware; write 1 and clear on the corresponding bit of DMA_INTFCLR. 0: Not generate 1: Generate
27,23,19, 15,11,7,3	TERRFLGx	R	Channel x Transfer Error Occur Flag (x=1..7) Indicate whether the transmission error interrupt (TE) is generated on the channel; these bits are set to 1 by hardware; write 1 and clear on the corresponding bit of DMA_INTFCLR. 0: Not generate 1: Generate

Field	Name	R/W	Description
31:28	Reserved		

10.6.2 DMA interrupt flag clear register (DMA_INTFCLR)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
24,20,16,12, 8,4,0	GINTCLRx	R/W	Channel x Global Interrupt Occur Flag Clear (x=1..7) Clear the corresponding GINTFLG, TCFLG, HTFLG and TERRFLG flags in the interrupt state register. 0: Invalid 1: Clear the GINTFLG flag
25,21, 17,13, 9,5,1	TCCLRx	R/W	Channel x Transfer Complete Clear (x=1..7) Clear the corresponding TCFLG flag in interrupt state register. 0: Invalid 1: Clear the TCFLG flag
26,22 18,14, 10,6,2	HTCLRx	R/W	Channel x Half Transfer Complete Clear (x=1..7) Clear the corresponding HTFLG flag in interrupt state register. 0: Invalid 1: Clear the HTFLG flag
27,23, 19,15, 11,7,3	TERRCLRx	R/W	Channel x Transfer Error Occur Clear (x=1..7) Clear the corresponding TERRFLG flag in interrupt state register. 0: Invalid 1: Clear the TERRFLG flag
31:28	Reserved		

10.6.3 DMA Channel x configuration register (DMA_CHCFGx) (x=1…7)

Offset address: 0x08+20 x (channel number–1)

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	CHEN	R/W	DMA Channel Enable 0: Disable 1: Enable
1	TCINTEN	R/W	All Transfer Complete Interrupt Enable) 0: Disable 1: Enable
2	HTINTEN	R/W	Half Transfer Complete Interrupt Enable 0: Disable 1: Enable
3	TERRINTEN	R/W	Transfer Error Occur Interrupt Enable 0: Disable 1: Enable
4	DIRCFG	R/W	Data Transfer Direction Configure 0: Read from peripheral to memory

Field	Name	R/W	Description
			1: Read from memory to peripheral
5	CIRMODE	R/W	Circular Mode Enable 0: Disable 1: Enable
6	PERIMODE	R/W	Peripheral Address Increment Mode Enable 0: Disable 1: Enable
7	MIMODE	R/W	Memory Address Increment Mode Enable 0: Disable 1: Enable
9:8	PERSIZE	R/W	Peripheral Data Size Configure 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved Note: It cannot be configured to 00 when I2C3 is used by user.
11:10	MEMSIZE	R/W	Memory Data Size Configure 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved Note: It cannot be configured to 00 when I2C3 is used by user.
13:12	CHPL	R/W	Channel Priority Level Configure 00: Low 01: Medium 10: High 11: Highest
14	M2MMODE	R/W	Memory to Memory Mode Enable 0: Disable 1: Enable
31:15	Reserved		

10.6.4 DMA Channel x transmission quantity register (DMA_CHNDATAx) (x=1…7)

Offset address: 0x0C+20 x (channel number–1)

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	NDATAT	R/W	Number of Data to Transfer Setup This register indicates the number of bytes to be transmitted. The number of data transmission ranges from 0 to 65535. This register can only be written when the channel is not working; once the channel is enabled, the register will be read-only, indicating the number of remaining bytes to be transmitted. The register will decrease after each DMA is transmitted; when the data transmission is completed, the register will change to 0, or when the channel is configured to auto reload mode, it will be automatically reloaded to the previously configured value; if the register is 0, data transmission will not occur regardless of whether the channel is turned on or not.
31:16	Reserved		

10.6.5 DMA Channel x peripheral address register (DMA_CHPADDRx) (x=1…7)

Offset address: 0x10+20 x (channel number–1)

Reset value: 0x0000 0000

This register cannot be written when the channel is turned on (CHEN=1 for DMA_CHCFGx).

Field	Name	R/W	Description
31:0	PERADDR	R/W	Peripheral Basic Address Setup When PERSIZE= '01' (16 bits) and PERADDR[0] bit is not used, it will be aligned with 16-bit address automatically during transmission. When PERSIZE= '10' (32 bits) and PERADDR[1:0] bit is not used, it will be aligned with 32-bit address automatically during transmission.

10.6.6 DMA Channel x memory address register (DMA_CHMADDRx) (x=1…7)

Offset address: 0x14+20 x (channel number–1)

Reset value: 0x0000 0000

This register cannot be written when the channel is turned on (CHEN=1 for DMA_CHCFGx).

Field	Name	R/W	Description
31:0	MEMADDR	R/W	Memory Basic Address Setup When MEMSIZE= '01' (16 bits) and MEMADDR[0] bit is not used, it will be aligned with 16-bit address automatically during transmission. When MEMSIZE= '10' (32 bits) and MEMADDR[1:0] bit is not used, it will be aligned with 32-bit address automatically during transmission.

11 Debug MCU (DBGMCU)

11.1 Full Name and Abbreviation Description of Terms

Table 47 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Frame Clock	FCLK
Serial Wire/JTAG Debug Port	SWJ-DP

11.2 Introduction

APM32F10x MCU series uses Arm® Cortex®-M3 core, and Arm® Cortex®-M3 core includes hardware debug module and supports complex debug operation. During debugging, the module can make the running core stop at breakpoint, and achieve the effect of querying the internal state of the core and the external state of the system, and after the query is completed, the core and peripheral operation can be restored to continue to execute the program.

Two debug interfaces are supported:

- Serial interface
- JTAG debug interface

Note: The hardware debug interface included in Arm® Cortex®-M3 core is subset of Arm CoreSight development tool set. Please refer to *Cortex®-M3 (Version r1p1) Technical Reference Manual (TRM)* and *CoreSight Development Tool Set (Version r1p0) TRM* for more information about debug function of Arm® Cortex®-M3 core.

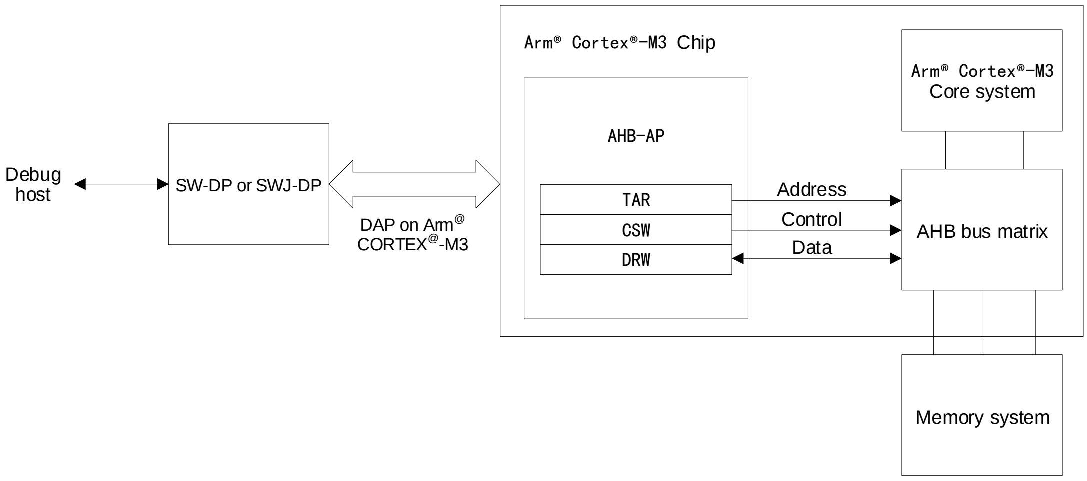
11.3 Main Characteristics

（1） Replace the core to access AHB bus matrix

（2） Flexible debug pin assignment

（3） MCU debug box (support low-power mode, control peripheral clock, etc.)

Figure 18 APM32F10xxx Level and Arm® Cortex®-M3 Level Debugging Block Diagram



11.4 Functional Description

11.4.1 Debug Pin Function Configuration

（1） Realize the on-line programming and debugging of the chip

（2） Using KEIL/IAR and other software to achieve on-line debugging, downloading and programming

（3） Flexible implementation of production of bus-line programmer

Table 48 Pin Function Configuration

SWJ-CFG[2:0]	Configured as dedicated pin for debugging	I/O port assignment of SWJ interface				
		PA13/ JTMS/ SWDIO	PA14/ JTCK/ SWCLK	PA15/ JTDI	PB3/ JTDO	PB4/ JNTRST
Others	Disable	Reserved				
100	Both JTAG-DP interface and SW-DP interface disabled					
010	JTAG-DP interface disabled, SW-DP interface enabled	Dedicated	Dedicated	Reserved		
001	All SWJ pins (JTAG-DP+SW-DP) Except JNTRST pin	Dedicated	Dedicated	Dedicated	Dedicated	Reserved
000	All SWJ pins (JTAG-DP+SW-DP) Reset state	Dedicated	Dedicated	Dedicated	Dedicated	Dedicated

Note: The items that cannot be tested in running mode can be observed and tested in detail

11.4.2 ID Code

11.4.2.1 MCU device ID code

APM32F MCU series incudes a MCU ID code. It can be accessed with JTAG or SW debug interface or user code.

11.4.2.2 Boundary scan TAP

JTAG ID code

The boundary scan TAP of APM32F MCU series integrates JTAG ID code. For APM32E103xCxE series products, its JTAG ID code is 0x06414b47

11.4.2.3 Arm® Cortex®-M3 TAP

Arm® Cortex®-M3 TAP has a JTAG ID code, which is 0x4BA00477.

11.4.2.4 Arm® Cortex®-M3 JEDEC-106 ID code

Arm® Cortex®-M3 has a JEDEC-106 ID code. It is located in 4KB ROM table in which the internal PPB bus address is 0xE00FF000_0xE00FFFFF.

11.5 Register Address Mapping

Table 49 Register Address Mapping

Register name	Description	Address
DBGMCU_IDCODE	Device ID register	0xE004 2000
DBGMCU_CFG	Debug MCU configuration register	0xE004 2004

11.6 Register Functional Description

11.6.1 Device ID register (DBGMCU_IDCODE)

Address: 0xE004 2000

Only support 32-bit access

Reset value: 0xXXXX XXXX, X=undefined bit

Field	Name	R/W	Description
11:0	EQR	R	Equipment Recognition For APM32F10x MCU series: APM32E103xCxE series products: 0x414; The debugger/programming tool identifies chips by QR (11:0).
15:12	Reserved		
31:16	WVR	R	Wafer Version Recognition This domain identifies wafer information

11.6.2 Debug MCU configuration register (DBGMCU_CFG)

This register can configure MCU in debug mode. It includes the counter supporting timer and watchdog, low-power mode, CAN communication and assignment tracking pin.

Address: 0xE004 2004

Only support 32-bit access

Reset value: 0x0000 0000 (not affected by system reset, only power-on reset)

Field	Name	R/W	Description
0	SLEEP_CLK_STS	R/W	Configure clock status when MCU is debugged in sleep mode 0: FCLK ON, HCLK OFF 1: FCLK ON, HCLK ON, provided by system clock
1	STOP_CLK_STS	R/W	Configure clock status when MCU is debugged in stop mode 0: FCLK OFF, HCLK OFF 1: FCLK ON, HCLK ON, provided by HSICLK
2	STANDBY_CLK_STS	R/W	Configure clock status when MCU is debugged in standby mode 0: FCLK OFF, HCLK OFF 1: FCLK ON, HCLK ON, provided by HSICLK
4:3	Reserved		
5	TRACE_IOEN	R/W	Trace Debug Pin Enable 0: Tracking debug pin disabled 1: Tracking debug pin enabled
7:6	TRACE_MODE	R/W	Trace Debug Pin Mode Configure Tracking debug pin mode can be configured only when TRACE_IOEN=1: 00: Asynchronous mode 01: Synchronous mode, the data length is 1 10: Synchronous mode, the data length is 2 11: Synchronous mode, the data length is 4
8	IWDT_STS	R/W	Configure Independent Watchdog Work Status When Core Is in Halted 0: Work normally 1: Stop working
9	WWDT_STS	R/W	Configure Window Watchdog Work Status When Core Is in Halted 0: Work normally 1: Stop working
13:10	TMRx_STS	R/W	ConfigureTimer Work Status When Core Is in Halted 0: Work normally 1: Stop working
14	CAN1_STS	R/W	Configure CAN1 Work Status When Core Is in Halted 0: Work normally 1: Freeze the receiver transmitter of CAN1

Field	Name	R/W	Description
15	I2C1_SMBUS_TIMEOUT_STS	R/W	Configure I2C1_SMBUS_TIMEOUT Work Status When Core Is in Halted 0: Work normally 1: Freeze the timeout mode of SMBUS
16	I2C2_SMBUS_TIMEOUT_STS	R/W	Configure I2C2_SMBUS_TIMEOUT Work Status When Core Is in Halted 0: Work normally 1: Freeze the timeout mode of SMBUS
20:17	TMRx_STS	R/W	ConfigureTimer Work Status When Core Is in Halted 0: When the core is halted, the clock can be provided to the counter of related timer, and the timer can output normally 1: When the core is halted, the clock will not be prvovided to the counter of the related timer and the timer output will be disabled
21	CAN2_STS	R/W	Configure CAN2 Work Status When Core Is in Halted 0: Work normally 1: Freeze the receiver transmitter of CAN2
31:22	Reserved		

12 General-Purpose Input/Output Pin (GPIO)

12.1 Full Name and Abbreviation Description of Terms

Table 50 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
P-channel Metal Oxide Semiconductor	P-MOS
N-channel Metal Oxide Semiconductor	N-MOS

12.2 Main Characteristics

GPIO port can configure the following functions through 32-bit configuration register (GPIOx_CFGLOW/GPIOx_CFGHIG) and two 32-bit data registers GPIOx_IDATA/GPIOx_ODATA):

（1） Input mode

- Analog input
- Floating input
- Pull-up input
- Pull-down input

（2） Output mode

- Push-pull output
- Open-drain output
- Configurable maximum output rate

（3） Multiplexing mode

- Push-pull multiplexing function
- Open-drain multiplexing function

（4） GPIO can be used as external interrupt/wake-up line

（5） Support locking I/O configuration function

12.3 Structure Block Diagram

Figure 19 GPIO Structure Block Diagram

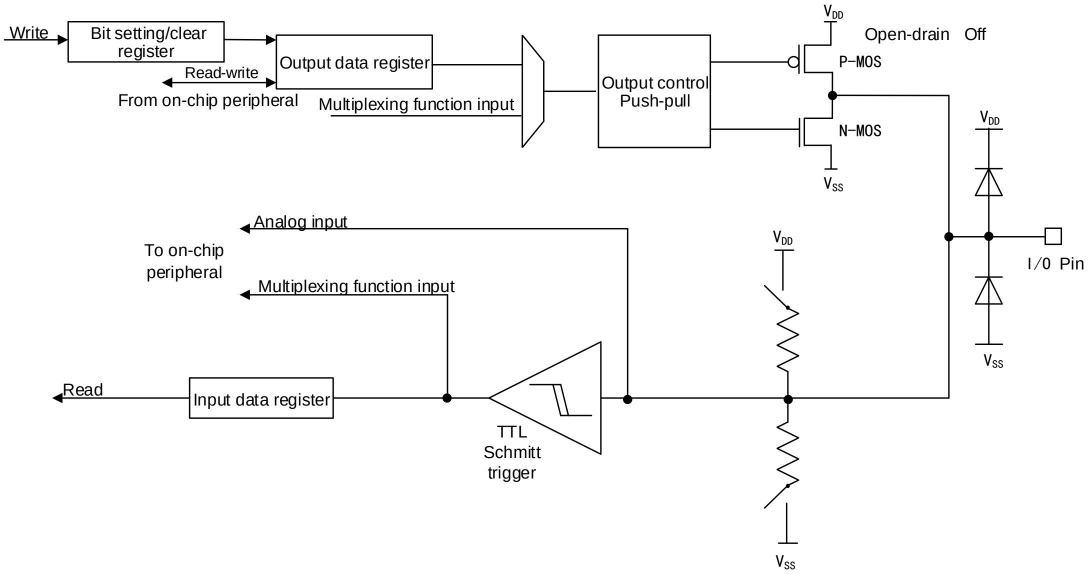
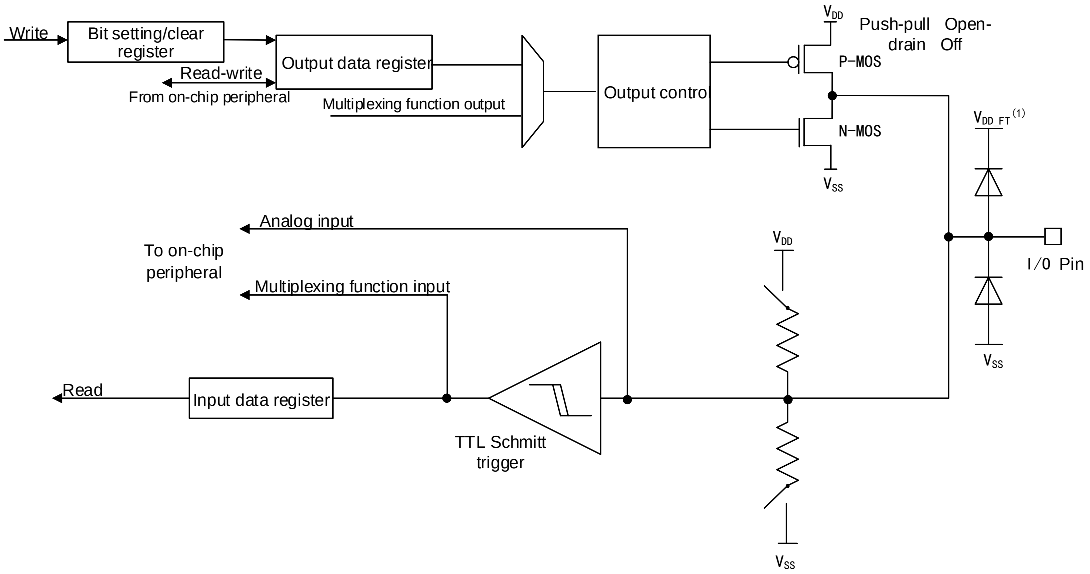


Figure 20 5V GPIO-compatible Structure Block Diagram



(1) V_{DD_FT} is different from V_{DD}, and V_{DD_FT} is special for FT GPIO pin.

12.4 Functional Description

Each pin of GPIO can be configured as pull-up, pull-down, floating and analog input, or push-pull/open-drain output input mode and multiplexing function through software. All GPIO interfaces have external interrupt capability.

12.4.1 IO status during Reset and just after Reset

If the multiplexing function is not enabled during and after GPIO reset, the I/O port will be configured as floating input mode, and in such case the pull-up/pull-down resistor is disabled in input mode. After reset, the JTAG pin is put in the input pull-up or pull-down mode, and the specific configuration is as follows:

- PA15: JTDI in pull-up mode
- PA14: JTCK in pull-down mode
- PA13: JTMS in pull-up mode
- PB4: JNTRST in pull-up mode

12.4.2 Input Mode

In the input mode, it can be set as pull-up, pull-down, floating and analog input.

When GPIO is configured as input mode, all GPIO pins have internal weak pull-up and weak pull-down resistors, which can be activated or disconnected.

Pull-up, pull-down, and floating modes

In (pull-up, pull-down, floating) input mode

- Schmitt trigger is opened,
- Disable output buffer
- Connect weak pull-up and pull-down resistors according to different input configurations
- The input data register GPIOx_IDATA captures the data on I/O pin in each APB2 clock cycle
- Read I/O state through the input data register GPIOx_IDATA

The initial level state of the floating input mode is uncertain and is easy to be disturbed by the outside; when connecting the equipment, it is determined by the external input level (except for the very high impedance).

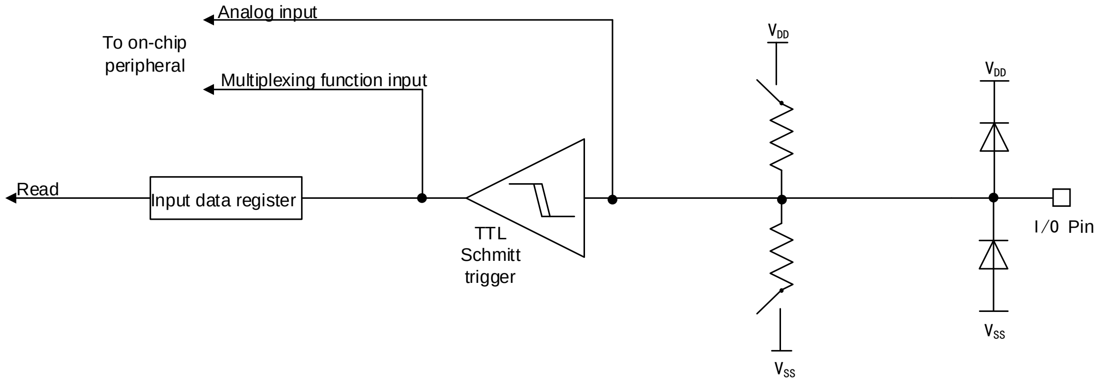
The initial level state of pull-up/pull-down input mode is high level if pull-up, and low level if pull-down; when connecting the equipment, it is determined by the external input level and load impedance.

Analog input mode

In analog input mode

- Disable output buffer
- The input of Schmitt trigger is disabled, and the output value of Schmitt trigger is forced to be 0
- Weak pull-up and pull-down resistors are disabled
- The value of port input state register is 0

Figure 21 Input Mode Structure



12.4.3 Output Mode

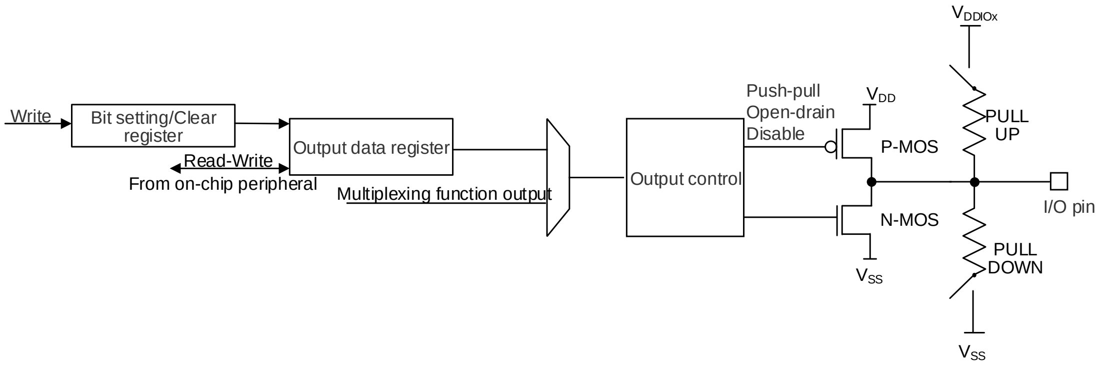
In the output mode, it can be set as push-pull output and open-drain output.

When GPIO is configured as the output pin, the output speed of the port can be configured and the output drive mode (push-pull / open-drain) can be selected.

In output mode

- Schmitt trigger is opened,
- Activate output buffer
- Weak pull-up and pull-down resistors are disabled
- Push-pull mode:
  - Double MOS transistor works by turns and the output data register can control the high and low level of I/O output
  - Read the finally written value through the output data register GPIOx_ODATA
- Open-drain mode:
  - Only N-MOS works, and the output data register can control I/O output high resistance state or low level
  - Read the actual I/O state through the input data register GPIOx_IDATA

Figure 22 I/O Structure in Output Mode



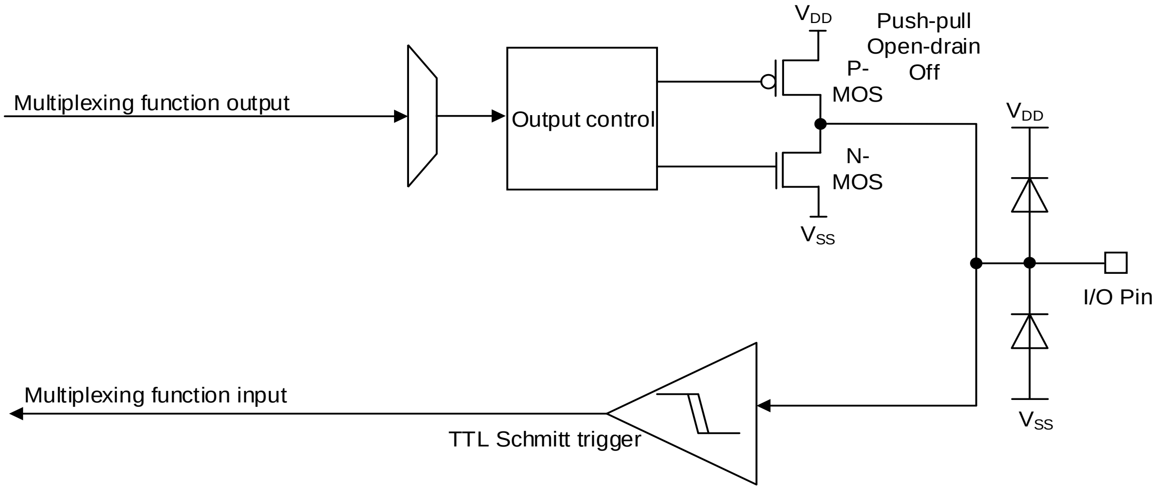
12.4.4 Multiplexing Mode

In multiplexing mode, it can be set as push-pull multiplexing and open-drain multiplexing

In push-pull/open-drain multiplexed mode

- Open the output buffer
- Output buffer is driven by peripheral
- Activate Schmitt trigger input
- Weak pull-up and pull-down resistors are disabled
- The data on the I/O pin is sampled in each APB2 clock cycle and stored in the port input state register
- In open-drain mode, the actual state of I/O can be read through input data register GPIOx_IDATA
- In push-pull mode, the last written value is read through output data register GPIOx_ODATA

Figure 23 I/O Structure in Multiplexing Mode



12.4.5 External Interrupt/Wake-up Line

All GPIO ports have external interrupt function. If you want to use external interrupt line, the port must be configured as input mode.

12.4.6 Bit Set and Bit Clear

Software does not need to disable interrupt when programming some bits of GPIOx_IDATA. (The function of changing one or more bits in APB2 write operation can be implemented by setting the bit to be changed in GPIOx_BSC 和 BSC register to 1.

12.4.7 GPIO Locking Function

Locking function can be used in power driver module. The locking mechanism of GPIO can protect the configuration of I/O port. I/O configuration can be locked by configuring the lock register (GPIOx_LOCK). When a port bit executes the locking program, the configuration of port bit cannot be modified before the next reset.

12.5 Register Address Mapping

Table 51 GPIO Register Address Mapping

Register name	Description	Offset address
GPIOx_CFGLOW	Low-8-bit port configuration low register	0x00
GPIOx_CFGHIG	High-8-bit port configuration high register	0x04
GPIOx_IDATA	Port input data register	0x08
GPIOx_ODATA	Port output data register	0x0C
GPIOx_BSC	Port bit set/clear register	0x10
GPIOx_BC	Port bit clear register	0x14
GPIOx_LOCK	Port configuration lock register	0x18

12.6 Register Functional Description

These peripheral registers must be operated by word (32 bits).

12.6.1 Low 8-bit port configuration register (GPIOx_CFGLOW) (x=A..G)

Offset address: 0x00

Reset value: 0x4444 4444

Field	Name	R/W	Description
29:28 25:24 21:20 17:16 13:12 9:8 5:4 1:0	MODEy[1:0]	R/W	Port x mode Configure (y=0…7) 00: Input mode (state after reset) 01: Output mode, the maximum output speed is 10MNz 10: Output mode, the maximum output speed is 2MNz 11: Output mode, the maximum output speed is 50MNz See the *Data Manual* for the definition of maximum output speed.
31:30 27:26 23:22 19:18 15:14 11:10 7:6 3:2	CFGy[1:0]	R/W	Port x Function Configure (y=0…7) The software configures corresponding I/O ports through these bits. In input (MODE[1:0]=00) mode 00: Analog input mode 01: Floating input mode (state after reset) 10: Pull-up/Pull-down input mode 11: Reserved In output mode (MODE[1:0]>00) 00: General push-pull output mode 01: General open-drain output mode 10: Push-pull output mode of multiplexing function

Field	Name	R/W	Description
			11: Open-drain output mode of multiplexing function

12.6.2 High 8-bit port configuration register (GPIOx_CFGLOW) (x=A..G)

Offset address: 0x04

Reset value: 0x4444 4444

Field	Name	R/W	Description
29:28 25:24 21:20 17:16 13:12 9:8 5:4 1:0	MODEy[1:0]	R/W	Port x mode Configure (y=8…15) The software configures corresponding I/O ports through these bits. 00: Input mode (state after reset) 01: Output mode, the maximum output speed is 10MNz 10: Output mode, the maximum output speed is 2MNz 11: Output mode, the maximum output speed is 50MNz See the *Data Manual* for the definition of maximum output speed.
31:30 27:26 23:22 19:18 15:14 11:10 7:6 3:2	CFGy[1:0]	R/W	Port x Function Configure (y=8…15) The software configures corresponding I/O ports through these bits. In input (MODE[1:0]=00) mode 00: Analog input mode 01: Floating input mode (state after reset) 10: Pull-up/Pull-down input mode 11: Reserved In output mode (MODE[1:0]>00) 00: General push-pull output mode 01: General open-drain output mode 10: Push-pull output mode of multiplexing function 11: Open-drain output mode of multiplexing function

12.6.3 Port input data register (GPIOx_IDATA) (x=A...G)

Offset address: 0x08

Reset value: 0x0000 XXXX

Field	Name	R/W	Description
15:0	IDATAy	R	Port input data (y=0…15) These bits are read-only and can be read out only in the form of word. 0: Output signal is at low level 1: Output signal is at high level
31:16	Reserved		

12.6.4 Port output data register (GPIOx_ODATA) (x=A…G)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	ODATAy	R/W	(Port output data) (y=0…15) These bits are readable and writable and can be operated only in the form of word. 0: Output low level 1: Output high level Note: For PIOx_BSC (x=A…E), each ODATAy bit can be set/cleared independently respectively.
31:16	Reserved		

12.6.5 Port bit setup/clear register (GPIOx_BSC) (x=A…G)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	BSy	W	Port x Set bit y (y=0…15) These bits are used to affect the corresponding ODATAy bits. 0: No effect 1: Set the corresponding ODATAy bits to 1 These bits are write-only and can be operated only in the form of word.
31:16	BCy	W	Port x Clear bit y (y=0…15) These bits are used to affect the corresponding ODATAy bits. 0: No effect 1: Corresponding ODATAy bit is cleared Note: BSy bit will work if the corresponding bits of both BSy and BCy are set. These bits are write-only and can be operated only in the form of word.

12.6.6 Port bit clear register (GPIOx_BC) (x=A…G)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	BCy	W	Port x Clear bit y (y=0…15) These bits are used to affect the corresponding ODATAy bits. 0: No effect 1: Corresponding ODATAy bit is cleared These bits are write-only and can be operated only in the form of word.
31:16	Reserved		

12.6.7 Port configuration lock register (GPIOx_LOCK) (x=A…G)

This register protects the configuration of GPIO from being modified by mistake during the running of the program. If the GPIO configuration is modified again, it can be modified only after the system is reset. When configuring GPIO configuration, it is necessary to write the specified sequence to the register to start the GPIO locking function.

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	LOCKy	R/W	Port x Pin y Lock bit y Configure (y=0…15) These bits decide whether to lock the port configuration. 0: The configuration of Port x Pin y is not locked 1: The configuration of Port x Pin y is locked These bits can be read and written, but can only be written when LOCKKEY=0.
16	LOCKKEY	R/W	Lock key value This bit determines whether the port configuration lock key bit is activated 0: Not activated 1: Activated; GPIOx_LOCK register is locked before the system is reset next time This bit can be read out at any time, and it can be written into the sequence modification through the lock key Write sequence of lock key: Write 1 Write 0 Write 1 Read 0 Read 1 Read 1 (the last read can be ignored, but can be used to confirm that the lock key has been activated.) Note: In the lock key value sequence, the value of LOCKy cannot be changed, and no error (sequence error, read error) can activate the lock protection.
31:17	Reserved		

13 Alternate Function Input and Output Pin (AFIO)

13.1 Introduction

In addition to the general I/O function, the I/O port can also be used as the interface of various peripheral functions. In order to make full use of the peripheral I/O pins of the product, the product supports the multiplexing function. It can not only realize multiple functions on the same pin (only one function can be realized at the same time), but also remap a certain function to other I/O (the originally supported function is no longer supported).

13.2 Functional Description

13.2.1 Alternate Function of I/O Pin

In order to make full use of peripheral I/O pins, some multiplexing functions can be remapped to other idle pins to maximize the utilization of pin resources.

Table 52 Corresponding Port Configuration of Multiplexing Function

Multiplexing function	**Configure for port bit configuration register**
Multiplexing input function	Configure as input mode and the input pin must be driven externally
Multiplexing output function	Configure as multiplexing function output mode
Bidirectional multiplexing function	Configure as multiplexing function output function, and the input driver is configured as floting input mode

Note:

(1) Through the GPIO controller programming, use the software to simulate the multiplexing function input pin, then the port is set to the multiplexing function output mode, and the pin is driven by software through the GPIO controller.

(2) When the multiplexing function is output, the pin is disconnected from the output register and connected with the output signal of the on-chip peripheral. If the peripheral is not activated after connection, the output of the pin will be uncertain.

13.2.1.1 Input mode configuration

When I/O port is used as input mode of multiplexing function, except that the weak pull-up and pull-down resistors are disabled, the port configuration is the same as that of general input function.

See corresponding chapters in GPIO for details of corresponding mode configuration.

13.2.1.2 Output mode configuration

When the I/O port is used as the output mode of multiplexing function (push-pull or open-drain), like the general output function, the output mode can be set as push-pull output and open-drain output, but the output buffer is driven by the signal of the built-in peripheral.

See corresponding chapters in GPIO for details of corresponding mode configuration.

Note: When software simulates multiplexing function input pin, the I/O port should be configured as multiplexing function output mode.

13.2.1.3 Bidirectional multiplexing function configuration

When bidirectional multiplexing function is used, the I/O port must be configured as multiplexing function output mode (push-pull or open-drain), while the input driver should be configured as floating input mode.

See corresponding chapters in GPIO for details of corresponding mode configuration.

13.2.2 Peripheral Pin Configuration

At this time, since the peripheral pin may have different functions, the I/O port configuration of the pin is different.

Table 53 TMR Pin Configuration

TMR pin	Configure	I/O port configuration
TMR1/8_CHx	Input capture channel x	Floating input
	Output compare channel x	Push-pull multiplexing output
TMR1/8_CHxN	Complementary output channel x	Push-pull multiplexing output
TMR1/8_BKIN	Braking input	Floating input
TMR1/8_ETR	External trigger clock input	Floating output
TMR2/3/4/5_CHx	Input capture channel x	Floating input
	Output compare channel x	Push-pull multiplexing output
TMR2/3/4/5_ETR	External trigger clock input	Floating input

Table 54 USAR Pin Configuration

USRAT pin	Configure	I/O port configuration
USRATx_TX	Full duplex mode	Push-pull multiplexing output
	Half duplex synchronous mode	Push-pull multiplexing output
USARTx_RX	Full duplex mode	Floating input or pull-up input
	Half duplex synchronous mode	Unused, can be used as GPIO

USRAT pin	Configure	I/O port configuration
USARTx_CK	Synchronous mode	Push-pull multiplexing output
USARTx_RTS	Hardware flow control	Push-pull multiplexing output
USARTx_CTS	Hardware flow control	Floating input or pull-up input

Table 55 SPI Pin Configuration

SPI pin	Configure	I/O port configuration
SPIx_SCK	Master mode	Push-pull multiplexing output
	Slave mode	Floating input
SPIx_MOSI	Full duplex mode/master mode	Push-pull multiplexing output
	Full duplex mode/slave mode	Floating input or pull-up input
	Simple bidirectional data cable/master mode	Push-pull multiplexing output
	Simple bidirectional data cable/slave mode	Unused, can be used as GPIO
SPIx_MISO	Full duplex mode/master mode	Floating input or pull-up input
	Full duplex mode/slave mode	Push-pull multiplexing output
	Simple bidirectional data cable/master mode	Unused, can be used as GPIO
	Simple bidirectional data cable/slave mode	Push-pull multiplexing output
SPIx_NSS	Hardware master/save mode	Floating input or pull-up input or pull-down input
	Hardware master mode/NSS output enable	Push-pull multiplexing output
	Software mode	Unused, can be used as GPIO

Table 56 I2S Pin Configuration

I2S pin	Configure	I/O port configuration
I2Sx_WS	Master mode	Push-pull multiplexing output
	Slave mode	Floating input
I2Sx_CK	Master mode	Push-pull multiplexing output
	Slave mode	Floating input
I2Sx_SD	Transmitter	Push-pull multiplexing output
	Receiver	Floating input or pull-up input or pull-down input
I2Sx_MCK	Master mode	Push-pull multiplexing output
	Slave mode	Unused, can be used as GPIO

Table 57 I2C Pin Configuration

I2C pin	Configure	I/O port configuration
I2Cx_SCL	I2C clock	Open-drain multiplexing output
I2Cx_SDA	I2C data	Open-drain multiplexing output

Table 58 BxCAN Pin Configuration

CAN pin	I/O port configuration
CAN_TX	Push-pull multiplexing output
CAN_RX	Floating input or pull-up input

Table 59 USBD Pin Configuration

USBD pin	I/O port configuration
USBD_DM/USBD_DP	After the USBD module is enabled, the pin will be automatically connected to the internal USBD transmitter receiver

Note: This table is only applicable to small-capacity, medium-capacity and large-capacity products.

Table 60 SDIO Pin Configuration

SDIO pin	I/O Configuration
SDIO_CK	Push-pull multiplexing output
SDIO_CMD	Push-pull multiplexing output
SDIO[D7:D0]	Push-pull multiplexing output

Table 61 ADC/DAC Pin Configuration

ADC/DAC pin	I/O port configuration
ADC/DAC	Analog input

Table 62 SMC Pin Configuration

SMC Pin	I/O port configuration
SMC_A[25:0]	Push-pull multiplexing output
SMC_D[15:0]	Push-pull multiplexing output
SMC_CK	Push-pull multiplexing output
SMC_NOE	Push-pull multiplexing output
SMC_NWE	Push-pull multiplexing output
SMC_NE[4:1]	Push-pull multiplexing output
SMC_NCE[3:2]	Push-pull multiplexing output
SMC_NCE4_1	Push-pull multiplexing output
SMC_NCE4_2	Push-pull multiplexing output
SMC_NWAIT	Floating input or pull-up input
SMC_CD	Floating input or pull-up input

SMC Pin	I/O port configuration
SMC_NIOS16	Floating input
SMC_INTR	Floating input
SMC_INT[3:2]	Floating input
SMC_NL	Push-pull multiplexing output
SMC_NBL[1:0]	Push-pull multiplexing output
SMC_NIORD	Push-pull multiplexing output
SMC_NIOWR	Push-pull multiplexing output
SMC_NREG	Push-pull multiplexing output

Table 63 DMC Pin Configuration

DMC Pin	I/O port configuration
DMC_A[15:0]	Push-pull multiplexing output
DMC_D[12:0]	Push-pull multiplexing output
DMC_CLK	Push-pull multiplexing output
DMC_CKE	Push-pull multiplexing output
DMC_NCS	Push-pull multiplexing output
DMC_NWE	Push-pull multiplexing output
DMC_NCAS	Push-pull multiplexing output
DMC_NRAS	Push-pull multiplexing output
DMC_UDQM	Push-pull multiplexing output
DMC_LDQM	Push-pull multiplexing output
DMC_BA0	Push-pull multiplexing output
DMC_BA1	Push-pull multiplexing output

13.2.3 Remapping Function Configuration

Generally, after the system reset, the pin will be given a default function; then if the user needs to multiplex other functions of the pin, as long as the peripheral is enabled, the multiplexing function can be activated. However, in addition that some peripheral functions need to be enabled, software programming is also needed to map the signal to the port, that is, assign the pin address, so that the peripheral function can be used in the pin.

The multiplexing function and remapping address table of pins are shown in the *Data Manual*.

13.2.3.1 OSC32_IN (OUT) pin is configured as GPIO

When not entering the standby mode or V_{DD} is not used for power supply, when

LSECLK oscillator is closed, the pin OSC32_IN/OSC32_OUT can be used as general I/O port PC14/PC15, namely, LSECLK function is prior to general I/O function.

13.2.3.2 OSC_IN (OUT) pin is configured as GPIO

In package products with less than 100 pins, the user can set AFIO_REMAP1/2 (multiplexing remapping and debug I/O configuration register) to realize the remapping of general I/O PD0/PD1 to external oscillator pin OSC_IN/OSC_OUT. Then PD0 and PD1 cannot be used to generate external interrupt time.

13.3 Register Address Mapping

Table 64 AFIO Register Address Mapping

Register name	Description	Offset address
AFIO_EVCTRL	Event control register	0x00
AFIO_REMAP1	Multiplexing remapping configuration register 1	0x04
AFIO_EINTSEL1	External interrupt configuration register 1	0x08
AFIO_EINTSEL2	External interrupt configuration register 2	0x0C
AFIO_EINTSEL3	External interrupt configuration register 3	0x10
AFIO_EINTSEL4	External interrupt configuration register 4	0x14
AFIO_REMAP2	Multiplexing remapping configuration register 2	0x18

13.4 Register Functional Description

For the register AFIO_EVCTRL, before read and write operation of AFIO_REMAP1/2 and AFIO_EINTSELx, AFIO clock shall be opened first. APB2 peripheral cock enable register (RCM_APB2CLKEN). These peripheral registers must be operated by word (32 bits).

13.4.1 Event control register (AFIO_EVCTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	PINSEL	R/W	Portx Piny Select (x=A…E) Pin y (y=0…15) Select the pin for outputting EVENTOUT signal of the core: 0000: Select Px0 0001: Select Px1 0010: Select Px2 0011: Select Px3 0100: Select Px4 0101: Select Px5 0110: Select Px6

Field	Name	R/W	Description
			0111: Select Px7 1000: Select Px8 1001: Select Px9 1010: Select Px10 1011: Select Px11 1100: Select Px12 1101: Select Px13 1110: Select Px14 1111: Select Px15
6:4	PORTSEL	R/W	Portx Select (x=A…G) Select the port for outputting EVENTOUT signal of the core: 000: Select PA 001: Select PB 010: Select PC 011: Select PD 100: Select PE 101: Select PF 110: Select PG 111: Reserved
7	EVOEN	R/W	Event Output Enable 0: Disable 1: Enable the EVENTOUT of the core to connect to Port x Pin y selected by PORTSEL[2:0] and PINSEL[3:0].
31:8	Reserved		

13.4.2 Multiplexing remapping configuration register 1 (AFIO_REMAP1)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	SPI1RMP	R/W	SPI1 Remap Configure 0: No remapping NSS—PA4, SCKPA5, MISO—PA6, MOSI—PA7 1: Remapping NSS—PA15, SCK—PB3, MISO—PB4, MOSI—PB5
1	I2C1RMP	R/W	I2C1 Remap Configure 0: No remapping SCL—PB6, SDA—PB7 1: Remapping SCL—PB8, SDA—PB9
2	USART1RMP	R/W	USART1 Remap Configure 0: No remapping TX—PA9, RX—PA10 1: Remapping TX—PB6, RX—PB7
3	USART2RMP	R/W	USART2 Remap Configure 0: No remapping

Field	Name	R/W	Description
			CTS—PA0, RTS—PA1, TX—PA2, RX—PA3, CK—PA4 1: Remapping CTS—PD3, RTS—PD4, TX—PD5, RX—PD6, CK—PD7
5:4	USART3RMP	R/W	USART3 Remap Configure 00: No remapping TX—PB10, RX—PB11, CK—PB12, CTS—PB13, RTS—PB14 01: Partial remapping TX—PC10, RX—PC11, CK—PC12, CTS—PB13, RTS—PB14 10: No effect 11: Complete remapping TX—PD8, RX—PD9, CK—PD10, CTS—PD11, RTS—PD12
7:6	TMR1RMP	R/W	TMR1 Remap Configure 00: No remapping ETR—PA12, CH1—PA8, CH2—PA9, CH3—PA10, CH4—PA11, BKIN—PB12, CH1N—PB13, CH2N—PB14, CH3N—PB15 01: Partial mapping ETR—PA12, CH1—PA8, CH2—PA9, CH3—PA10, CH4—PA11, BKIN—PA6, CH1N—PA7, CH2N—PB0, CH3N—PB1 10: No effect 11: Complete mapping ETR—PE7, CH1—PE9, CH2—PE11, CH3—PE13, CH4—PE14, BKIN—PE15, CH1N—PE8, CH2N—PE10, CH3N—PE12
9:8	TMR2RMP	R/W	TMR2 Remap Configure 00: No remapping CH1/ETR—PA0, CH2—PA1, CH3—PA2, CH4—PA3 01: Partial remapping CH1/ETR—PA15, CH2—PB3, CH3—PA2, CH4—PA3 10: Partial remapping CH1/ETR—PA0, CH2—PA1, CH3—PB10, CH4—PB11 11: Complete remapping CH1/ETR—PA15, CH2—PB3, CH3—PB10, CH4—PB11
11:10	TMR3RMP	R/W	TMR3 Remap Configure 00: No remapping CH1—PA6, CH2—PA7, CH3—PB0, CH4—PB1 01: No effect 10: Partial mapping CH1—PB4, CH2—PB5, CH3—PB0, CH4—PB1 11: Complete mapping CH1—PC6, CH2—PC7, CH3—PC8, CH4—PC9 Note: Remapping does not affect TMR3_ETR on PD2.
12	TMR4RMP	R/W	TMR4 Remap Configure 0: No remapping TMR4_CH1—PB6, TMR4_CH2—PB7, TMR4_CH3—PB8, TMR4_CH4—PB9 1: Complete mapping TMR4_CH1—PD12, TMR4_CH2—PD13, TMR4_CH3—PD14, TMR4_CH4—PD15

Field	Name	R/W	Description
			Note: Remapping does not affect TMR4_ETR on PE0.
14:13	CAN1RMP	R/W	CAN1 Remap Configure 00: CAN1_RX mapped to PA11, CAN1_TX mapped to PA12 01: No effect 10: CAN1_RX mapped to PB8, CAN1_TX mapped to PB9 11: CAN1_RX mapped to PD0, CAN1_TX mapped to PD1
15	PD01RMP	R/W	Port D0/Port D1 mapping on OSC_IN/OSC_OUT Configure This function can only be used for Pin 64 package (PD0 and PD1 appear on Pin 100 package without remapping). 0: No remapping for PD0 and PD1 1: PD0 mapped to OSC_IN, PD1 mapped to OSC_OUT When the main oscillator HSECLK is not used (the system runs in internal 8MHz resistance-capacitance oscillator), PD0 and PD1 can be mapped to OSC_IN and OSC_OUT pins.
16	TMR5CH4IRMP	R/W	TMR5CH4 Interrupt Remap This bit can be set to 1 or cleared by software. It controls internal mapping of TMR5 Channel 4. 0: TMR5_CH4 is connected to PA3; 1L LIRC internal oscillator is connected to TMR5_CH4 for calibration of LIRC.
17	ADC1_ETRGINJC_RMP	R/W	ADC1 External Trigger Injected Conversion Remapping Configure 0: External trigger of ADC1 injected conversion is connected to EINT15 1: External trigger of ADC1 injected conversion is connected to TMR8_CH4
18	ADC1_ETRGREGC_RMP	R/W	ADC1 External Trigger Regular Conversion Remapping Configure 0: External trigger of ADC1 regular conversion is connected to EINT11 1: External trigger of ADC1 regular conversion is connected to TMR8_TRGO
19	ADC2_ETRGINJC_RMP	R/W	ADC2 External Trigger Injected Conversion Remapping Configure 0: External trigger of ADC2 injected conversion is connected to EINT15 1: External trigger of ADC2 injected conversion is connected to TMR8_CH4
20	ADC2_ETRGREGC_RMP	R/W	ADC2 External Trigger Regular Conversion Remapping Configure 0: External trigger of ADC2 regular conversion is connected to EINT11 1: External trigger of ADC2 regular conversion is connected to TMR8_TRGO
21	Reserved		
22	CAN2RMP	R/W	CAN2 Remap Configure 0: No remapping CAN2_RX—PB12, CAN2_TX—PB13 1: Remapping CAN2_RX—PB5, CAN2_TX—PB6
23	Reserved		

Field	Name	R/W	Description
26:24	SWJCFG	W	Serial Wire JTAG Configure Configure SWJ and tracking multiplexing function I/O as debugging I/O or normal I/O, applicable when GPIO is not enough. These bits can only be written by software (read these bits and undefined values will be returned). SWJ (serial line JTAG) supports JTAG or SWD to access the debug port of Cortex. The default state after system reset is enabled SWJ without tracking function. 000: Complete SWJ (JTAG-DP+SW-DP) 001: Complete SWJ (JTAG-DP+SW-DP) but without NJTRST 010: JTAG-DP disabled, SW-DP enabled 100: JTAG-DP disabled, SW-DP disabled Others: Reserved
31:27	Reserved		

13.4.3 External interrupt configuration register 1 (AFIO_EINTSEL1)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	EINTx[3:0]	R/W	EINTx Input Source Select 0000: PA[x] pin 0001: PB[x] pin 0010: PC[x] pin 0011: PD[x] pin 0100: PE[x] pin 0101: PF[x] pin 0110: PG[x] pin Others: Reserved
31:16	Reserved		

13.4.4 External interrupt configuration register 2 (AFIO_EINTSEL2)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	EINTx[3:0]	R/W	EINTx Input Source Select (x=4…7) 0000: PA[x] pin 0001: PB[x] pin 0010: PC[x] pin 0011: PD[x] pin 0100: PE[x] pin 0101: PF[x] pin 0110: PG[x] pin Others: Reserved
31:16	Reserved		

13.4.5 External interrupt configuration register 3 (AFIO_EINTSEL3)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	EINTx[3:0]	R/W	EINTx Input Source Select (x=8…11) 0000: PA[x] pin 0001: PB[x] pin 0010: PC[x] pin 0011: PD[x] pin 0100: PE[x] pin 0101: PF[x] pin 0110: PG[x] pin Others: Reserved
31:16	Reserved		

13.4.6 External interrupt configuration register 4 (AFIO_EINTSEL4)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	EINTx[3:0]	R/W	EINTx Input Source Select (x=12…15) 0000: PA[x] pin 0001: PB[x] pin 0010: PC[x] pin 0011: PD[x] pin 0100: PE[x] pin 0101: PF[x] pin 0110: PG[x] pin Others: Reserved
31:16	Reserved		

13.4.7 Multiplexing remapping configuration register 2 (AFIO_REMAP2)

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
9:0	Reserved		
10	EMMCNADV	R/W	NADV Connect/Disconnect The bit is controlled and modified by software. 0: NADV flag is connected to output (there is error) 1: NADV flag is not connected to output, and I/O pin can be used by other peripherals.
31:11	Reserved		

14 Timer Overview

14.1 Full Name and Abbreviation Description of Terms

Table 65 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Timer	TMR
Update	U
Request	R
Event	EV
Capture	C
Compare	C
Length	LEN

14.2 Timer Category and Main Difference

In this series of products, there are three types of timers: advanced timer, general-purpose timer and basic timer (watchdog timer and system tick timer are described in other chapters).

The advanced timer includes the functions of general-purpose timer and basic timer. The advanced timer has four capture/compare channels, supports timing function, input capture and output compare function, braking and complementary output function, and is a 16-bit timer that can count up/down.

The function of general-purpose timer is simpler than that of advanced timer. The main differences are the total number of channels, the number of complementary output channel groups and the braking function.

The basic timer is a timer that can only realize timing function and has no external interface.

The main differences of timers included in the products are shown in the table below:

Table 66 Main Differences among Timers Included in the Products

Item	Specific content/Category	Advanced timer		General-purpose timer	Basic timer	
Name	—	TMR1	TMR8	TMR2/3/4/5	TMR6	TMR7
Timebase unit	Counter	16 bits		16 bits	16 bits	
	Prescaler	16 bits		16 bits	16 bits	

Item	Specific content/Category	Advanced timer	General-purpose timer	Basic timer
	Count mode	Up Down Center-aligned	Up Down Center-aligned	Up
Channel	Input channel	4	4	0
	Capture/Compare channel	4	4	0
	Output channel	7	4	0
	Complementary output channel	3 groups	0	0
Function	General DMA request	OK	OK	OK
	PWM mode	Yes	Yes	None
	Single-pulse mode	Yes	Yes	None
	Forced output mode	Yes	Yes	None
	Dead-time insertion	Yes	None	None

Timer term

Table 67 Definitions and Terms of Pins

Name	Description
TMRx_ETR	External trigger signal of Timer x
TMRx_CH1、TMRx_CH2、TMRx_CH3、TMRx_CH4	Channel 1/2/3/4 of Timer x
TMRx_ChyN	Complementary output channel y of Timer x
TMRx_BKIN	Braking signal of Timer x

Table 68 Definitions and Terms of Internal Signals

Name	Description
ETR	TMRx_ETR external trigger signal
ETRF	External trigger filter
ETRP	External trigger prescaler
-	
ITR, ITR0, ITR1	Internal trigger
TRGI	Clock/Trigger/Slave mode controller trigger input
TIF_ED	Timer input filter edge detection

Name	Description
-	
CK_PSC	Prescaler clock
CK_CNT	Counter clock
PSC	Prescaler
CNT	Counter
AUTORLD	Autoload register
-	
TIx, TI1	Timer input
TIxF, TI1F,	Timer input filter
TI1_ED	Timer input edge detection
TIxFPx, TI1FP1	Timer input filter polarity
ICx, IC1	Input capture
ICxPS, IC1PS	Input capture prescaler
TRC	Trigger capture
BRK	Braking signal
-	
OCx, OC1	Timer output coparison channel
OCxREF, OC1REF	Output compare reference signal
-	
TGI	Trigger interrupt
BI	Braking interrupt
CCxI, CC1I	Capture/Compare interrupt
UEV	Update event
UIFLG	Update interrupt flag

15 Advanced Timer (TMR1/8)

15.1 Introduction

The advanced timer takes the time base unit as the core, and has the functions of input capture, output compare and braking input, including a 16-bit automatic loading counter. Compared with other timers, the advanced timer supports complementary output, repeat count and programmable dead-time insertion function, and is more suitable for motor control.

15.2 Main Characteristics

（1） Timebase unit

- Counter: 16-bit counter, count-up, count-down and center-aligned count
- Prescaler: 16-bit programmable prescaler
- Repeat counter: 16-bit repeat counter
- Auto reloading function

（2） Clock source selection

- Internal clock
- External input
- External trigger
- Internal trigger

（3） Input capture function

- Counting function
- PWM input mode (measurement of pulse width, frequency and duty cycle)
- Encoder interface mode

（4） Output compare function

- PWM output mode
- Forced output mode
- Single-pulse mode
- Complementary output and dead-time insertion

（5） Timing function

（6） Braking function

（7） Master/Slave mode controller of timer

- Timers can be synchronized and cascaded
- Support multiple slave modes and synchronization signals

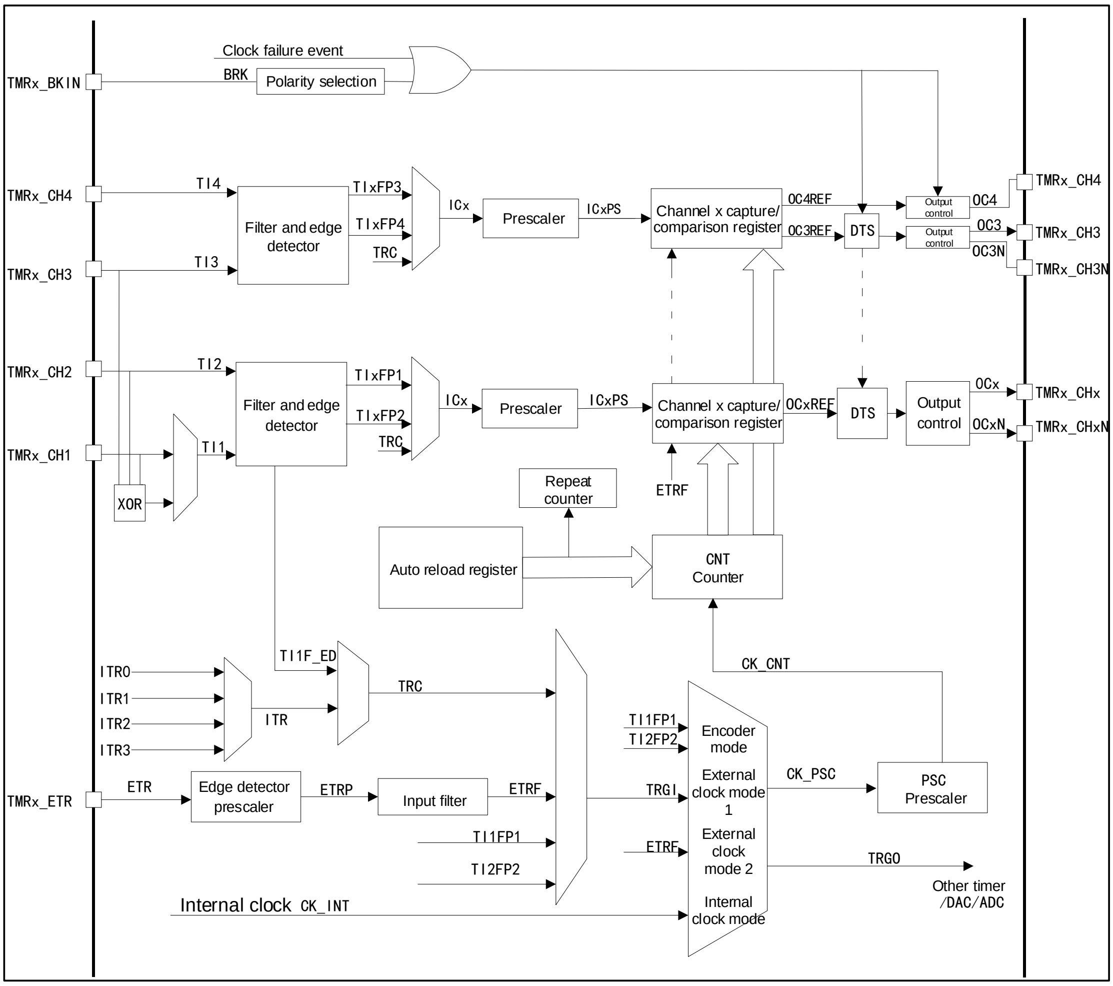
（8） Interrupt output and DMA request event

- Update event (counter overrun/underrun, counter initialization)

- Trigger event (counter start, stop, internal/external trigger)
- Capture/Compare event
- Braking signal input event

15.3 Structure Block Diagram

Figure 24 Advanced Timer Structure Block Diagram



15.4 Functional Description

15.4.1 Clock Source Selection

The advanced timer has four clock sources

Internal clock

It is TMRx_CLK from RCM, namely the driving clock of the timer; when the slave mode controller is disabled, the clock source CK_PSC of the prescaler is driven by the internal clock CK_INT.

External clock mode 1

The trigger signal generated from the input channel TI1/2/3/4 of the timer after polarity selection and filtering is connected to the slave mode controller to control the work of the counter. Besides, the pulse signal generated by the input of Channel 1 after double-edge detection of the rising edge and the falling edge is logically equal or the future signal is TI1F_ED signal, namely double-edge signal of TIF_ED. Specially the PWM input can only be input by TI1/2.

External clock mode 2

After polarity selection, frequency division and filtering, the signal from external trigger interface (ETR) is connected to slave mode controller through trigger input selector to control the work of counter.

Internal trigger input

The timer is set to work in slave mode, and the clock source is the output signal of other timers. At this time, the clock source has no filtering, and the synchronization or cascading between timers can be realized. The master mode timer can reset, start, stop or provide clock for the slave mode timer.

15.4.2 Timebase Unit

The time base unit in the advanced timer contains four registers

- Counter register (CNT) 16 bits
- Auto reload register (AUTORLD) 16 bits
- Prescaler (PSC) 16 bits
- Repetition count register (REPCNT) 8 bits

Repetition register is unique to advanced timer.

Counter CNT

There are three counting modes for the counter in the advanced timer

- Count-up mode
- Count-down mode
- Center-aligned mode

Count-up mode

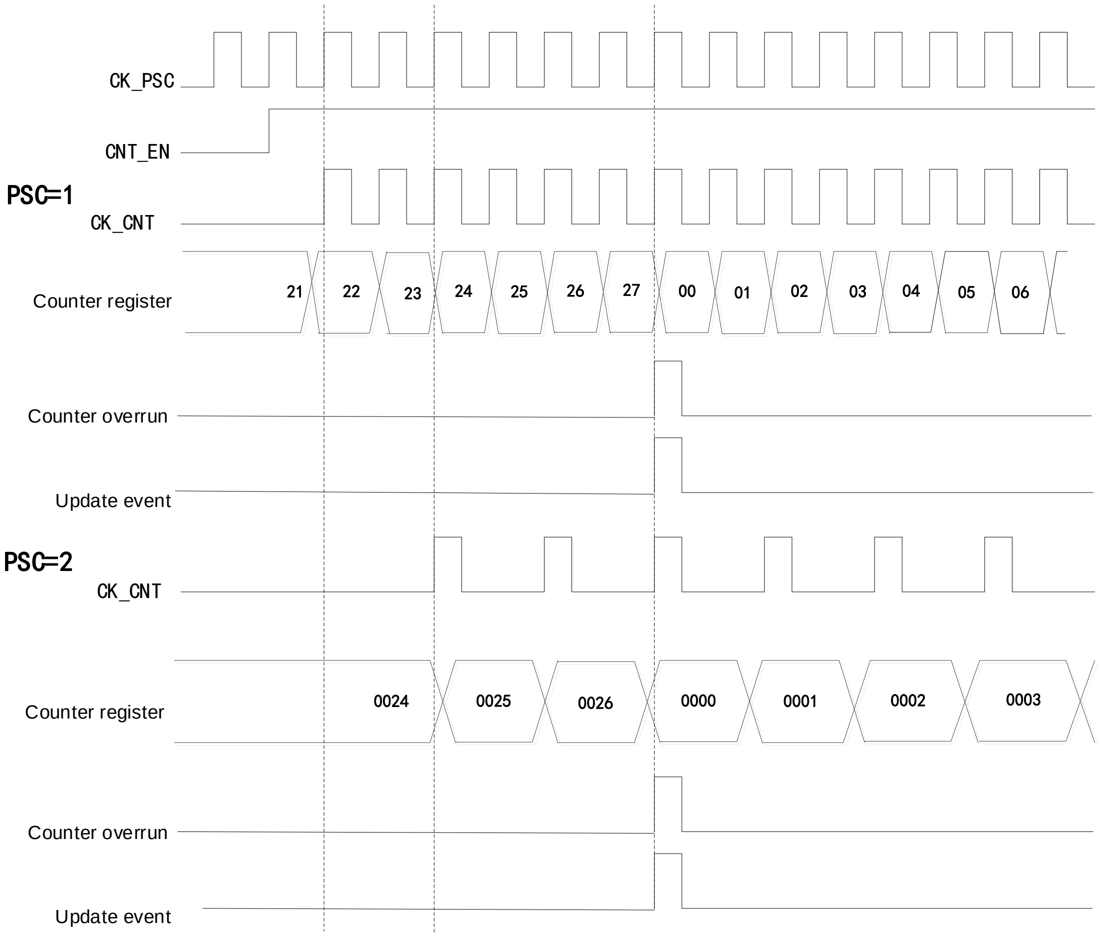
Set to the count-up mode by CNTDIR bit of configuration control register (TMRx_CTRL1).

When the counter is in count-up mode, the counter will count up from 0; every time a pulse is generated, the counter will increase by 1 and when the value of the counter (TMRx_CNT) is equal to the value of the auto reload (TMRx_AUTORLD), the counter will start to count again from 0, a count-up overrun event will be generated, and the value of the auto reload (TMRx_AUTORLD) is written in advance.

When the counter overruns, an update event will be generated. At this time, the repeat count shadow register, the automatic reload shadow register and the prescaler buffer will be updated. The update event can be disabled by UD bit of configuration control register TMRx_CTRL1.

The figure below is Timing Diagram when Division Factor is 1 or 2 in Count-up Mode

Figure 25 Timing Diagram when Division Factor is 1 or 2 in Count-up Mode



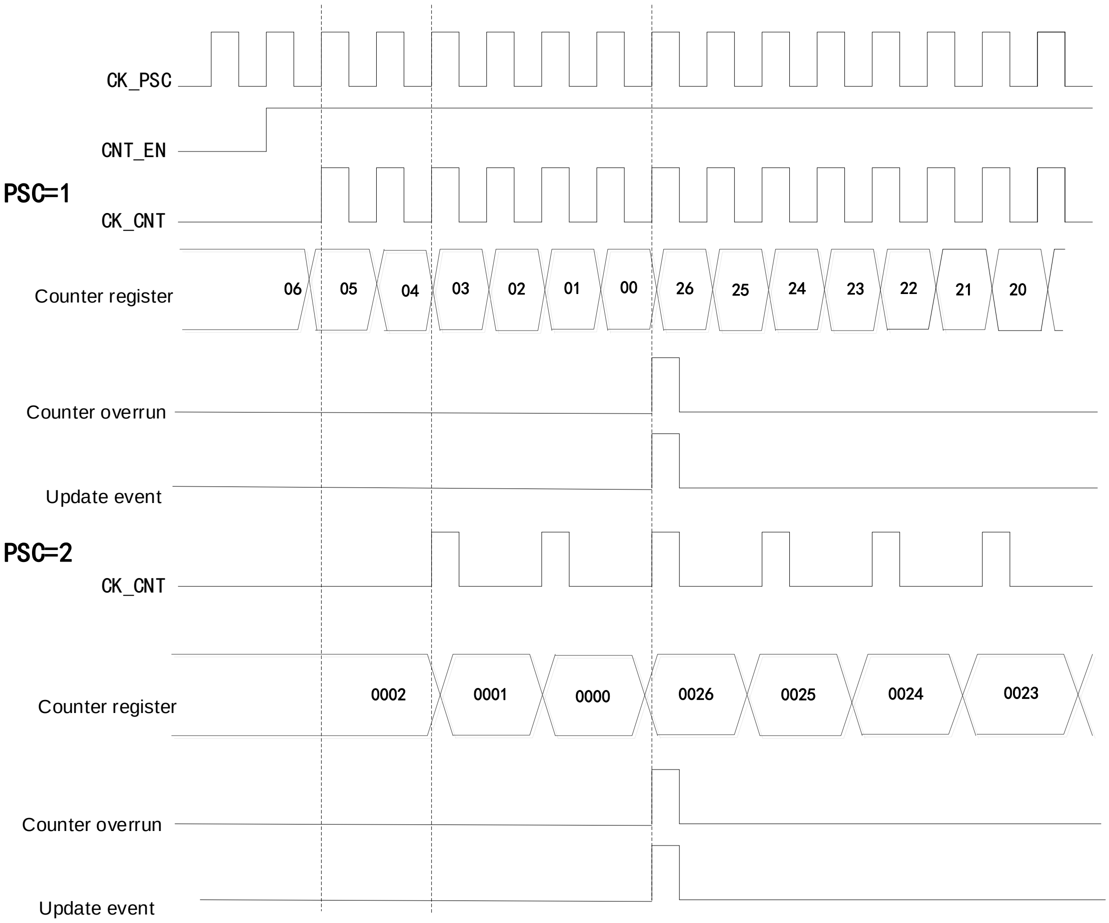
Count-down mode

Set to the count-down mode by CNTDIR bit of configuration control register (TMRx_CTRL1).

When the counter is in count-down mode, the counter will start to count down from the value of the auto reload (TMRx_AUTORLD); every time a pulse is generated, the counter will decrease by 1 and when it becomes 0, the counter will start to count again from (TMRx_AUTORLD), meanwhile, a count-down overrun event will be generated, and the value of the auto reload (TMRx_AUTORLD) is written in advance.

When the counter overruns, an update event will be generated. At this time, the repeat count shadow register, the automatic reload shadow register and the prescaler buffer will be updated. The update event can be disabled by configuring the UD bit of the TMRx_CTRL1 register.

Figure 26 Timing Diagram when Division Factor is 1 or 2 in Count-down Mode

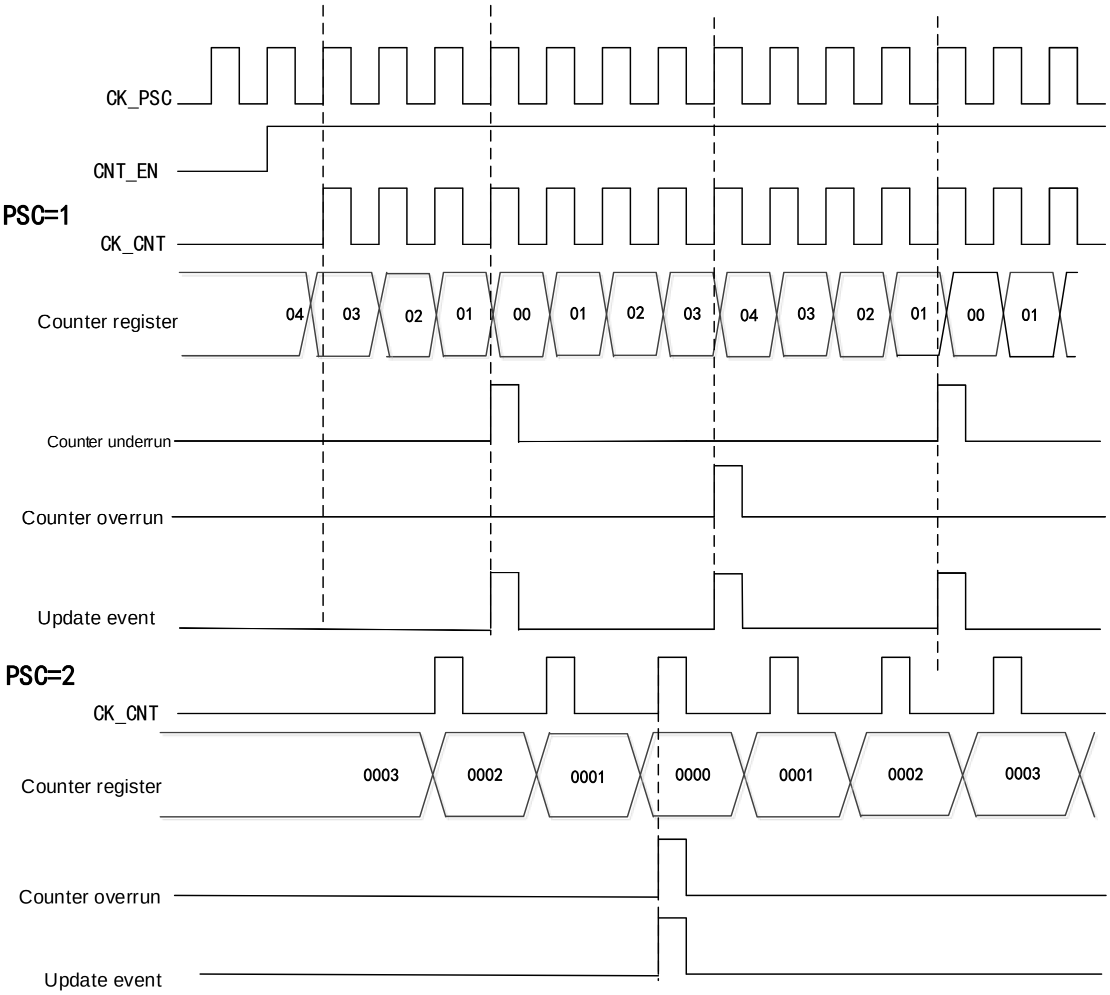


Center-aligned mode

Set to the center-aligned mode by CNTDIR bit of configuration control register (TMRx_CTRL1).

When the counter is in center-aligned mode, the counter counts up from 0 to the value of auto reload (TMRx_AUTORLD), then counts down to 0 from the value of the auto reload (TMRx_AUTORLD), which will repeat; in counting up, when the counter value is (AUTORLD-1), a counter overrun event will be generated; in counting down, when the counter value is 1, a counter underrun event will be generated.

Figure27 Timing Diagram when Division Factor is 1 or 2 in Center-aligned Mode



Repeat counter REPCNT

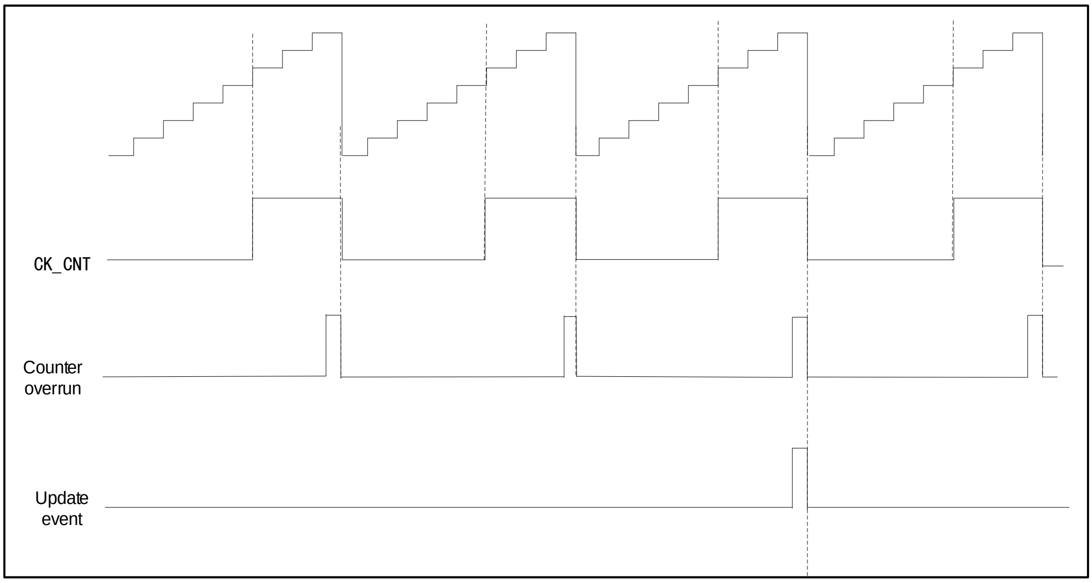
There is no repeat counter REPCNT in the basic/general-purpose timer, which means that when the overrun event or underrun event occurs in the basic/general-purpose timer, an update event will be generated directly; while in the advanced timer, because of the existence of the repeat counter, when an overrun/unerrrun event occurs to the advanced timer, the update event will be generated only when the value of the repeat counter is 0.

For example, if the advanced timer needs to generate an update event when an overrun/underrun event occurs, the value of the repeat counter should be set to 0.

If the repeat counter function is used in the count-up mode, every time the counter counts up to AUTORLD, an overrun event will occur. At this time, the value of the repeat counter will be decreased by 1, and an update event will be generated until the value of the repeat counter is 0.

That is, when N+1 (N is the value of repeat counter) overrun/underrun events occur, an update event will be generated.

Figure 28 Timing Diagram when Setting REPCNT=2 in Count-up Mode



Prescaler PSC

The prescaler is 16 bits and programmable, and it can divide the clock frequency of the counter to any value between 1 and 65536 (controlled by TMRx_PSC register), and after frequency division, the clock will drive the counter CNT to count. The prescaler has a buffer, which can be changed during running.

15.4.3 Input Capture

Input capture channel

The advanced timer has four independent capture/compare channels, each of which is surrounded by a capture/compare register.

In the input capture, the measured signal will enter from the external pin T1/2/3/4 of the timer, first pass through the edge detector and input filter, and then into the capture channel. Each capture channel has a corresponding capture register. When the capture occurs, the value of the counter CNT will be latched in the capture register CCx. Before entering the capture register, the signal will pass through the prescaler, which is used to set how many events to capture at a time.

Input capture application

Input capture is used to capture external events, and can give the time flag to indicate the occurrence time of the event and measure the pulse jump edge events (measure the frequency or pulse width), for example, if the selected edge appears on the input pin, the TMRx_CCx register will capture the current value of the counter and the CCxIFLG bit of the state register TMRx_STS will be set to 1; if CCxIEN=1, an interrupt will be generated.

In capture mode, the timing, frequency, period and duty cycle of a waveform can be measured. In the input capture mode, the edge selection is set to rising edge detection. When the rising edge appears on the capture channel, the first capture occurs, at this time, the value of the counter CNT will be latched in the capture register CCx; at the same time, it will enter the capture interrupt, a capture will be recorded in the interrupt service program and the value will be recorded. When the next rising edge is detected, the second capture occurs, the value of counter CNT will be latched in capture register CCx again, at this time, it will enter the capture interrupt again, the value of capture register will be read, and the cycle of this pulse signal will be obtained through capture.

15.4.4 Output Compare

There are eight modes of output compare: freeze, channel x is valid level when matching, channel x is invalid level when matching, flip, force is invalid, force is valid, PWM1 and PWM2 modes, which are configured by OCxMOD bit in TMRx_CCMx register and can control the waveform of output signal in output compare mode.

Output compare application

In the output compare mode, the position, polarity, frequency and time of the pulse generated by the timer can be controlled.

When the value of the counter is equal to that of the capture/compare register, the channel output can be set as high level, low level or flip by configuring the OCxMOD bit in TMRx_CCMx register and the CCxPOL bit in the output polarity TMRx_CCEN register.

When CCxIFLG=1 in TMRx_STS register, if CCxIEN=1 in TMRx_DIEN register, an interrupt will be generated; if CCDSEL=1 in TMRx_CTRL2 register, DMA request will be generated.

15.4.5 PWM Output Mode

PWM mode is an adjustable pulse signal output by the timer. The pulse width of the signal is determined by the value of the compare register CCx, and the cycle is determined by the value of the auto reload AUTORLD.

PWM output mode contains PWM mode 1 and PWM mode 2; PWM mode 1 and PWM mode 2 are divided into count-up, count-down and edge alignment counting; in PWM mode 1, if the value of the counter CNT is less than the value of the compare register CCx, the output level will be valid; otherwise, it will be invalid.

Set the timing diagram in PWM1 mode when CCx=5, AUTORLD=7

Figure 29 PWM1 Count-up Mode Timing Diagram

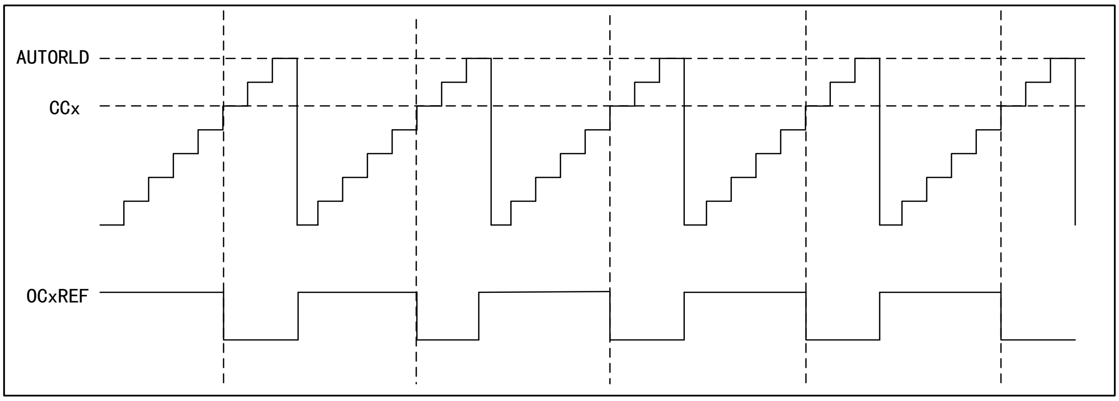


Figure 30 PWM1 Count-down Mode Timing Diagram

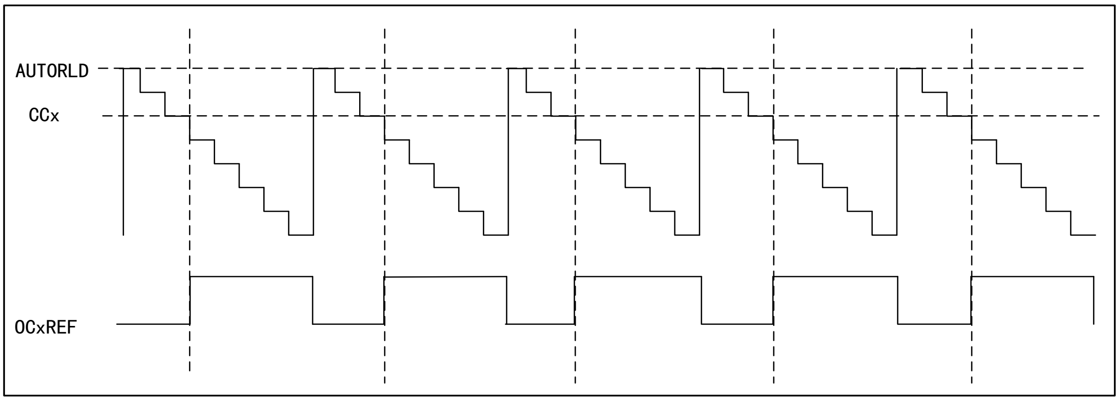
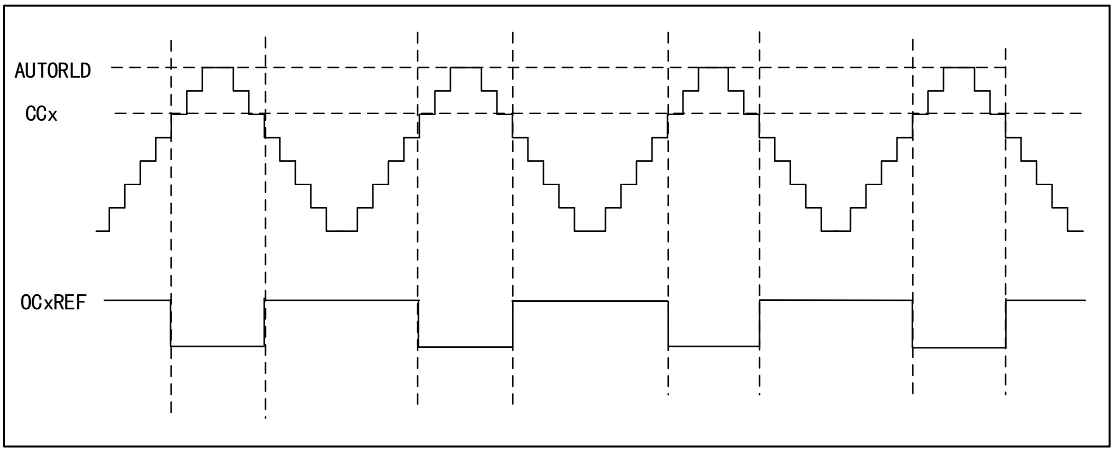


Figure 31 PWM1 Center-aligned Mode Timing Diagram



In PWM mode 2, if the value of the counter CNT is less than that of the compare register CCx, the output level will be invalid; otherwise, it will be valid.

Set the timing diagram in PWM mode 2 when CCx=5, AUTORLD=7

Figure 32 PWM2 Count-up Mode Timing Diagram

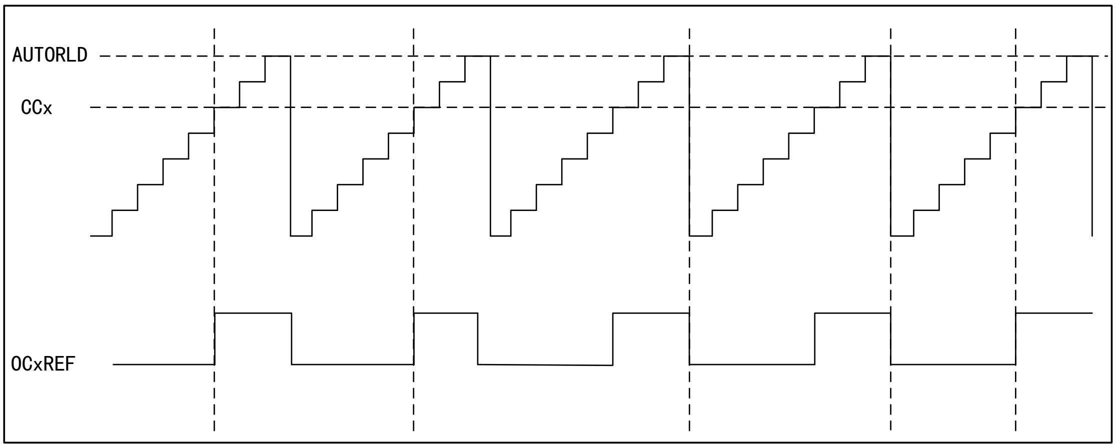


Figure 33 PWM2 Count-down Mode Timing Diagram

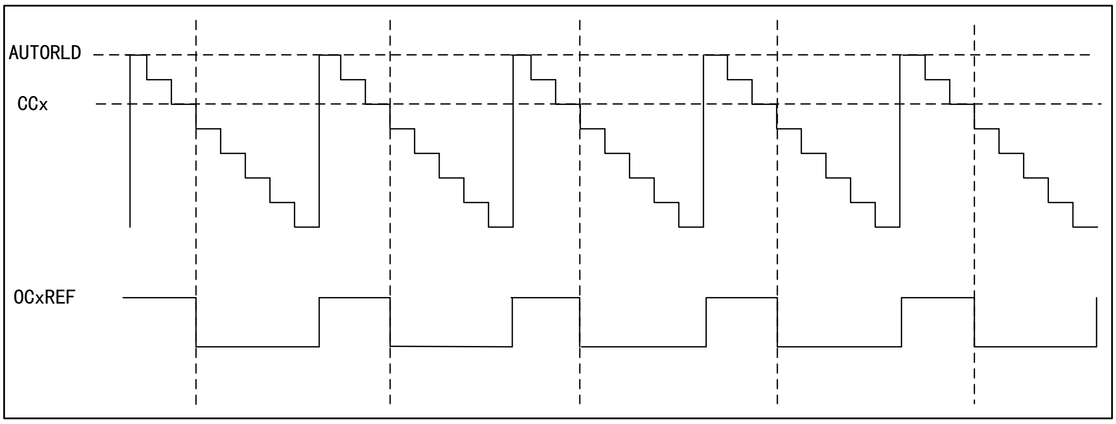
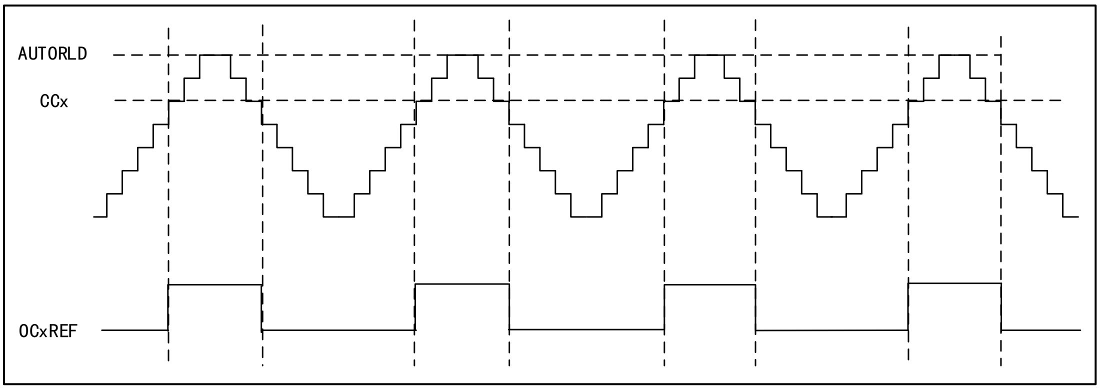


Figure 34 PWM2 Center-aligned Mode Timing Diagram



15.4.6 PWM Input Mode

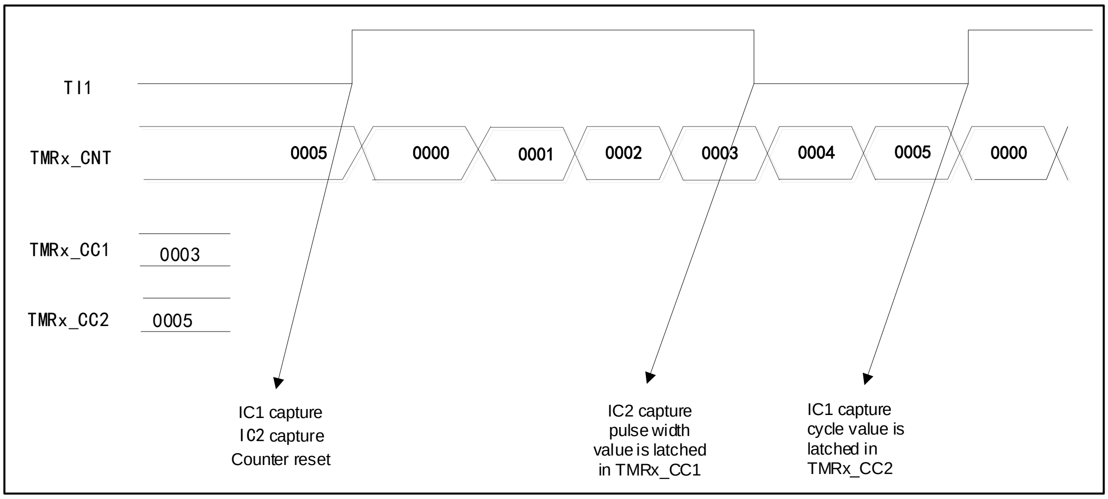
PWM input mode is a particular case of input capture.

In PWM input mode, as only TI1FP1 and TI1FP2 are connected to the slave mode controller, input can be performed only through the channels TMRx_CH1 and TMRx_CH2, which need to occupy the capure registers of CH1 and CH2.

In the PWM input mode, the PWM signal enters from TMRx_CH1, and the signal will be divided into two channels, one can measure the cycle and the other can measure the duty cycle. In the configuration, it is only required to set the polarity of one channel, and the other will be automatically configured with the opposite polarity.

In this mode, the slave mode controller should be configured as the reset mode (SMFSEL bit of TMRx_SMCTRL register)

Figure 35 PWM Input Mode Timing Diagram



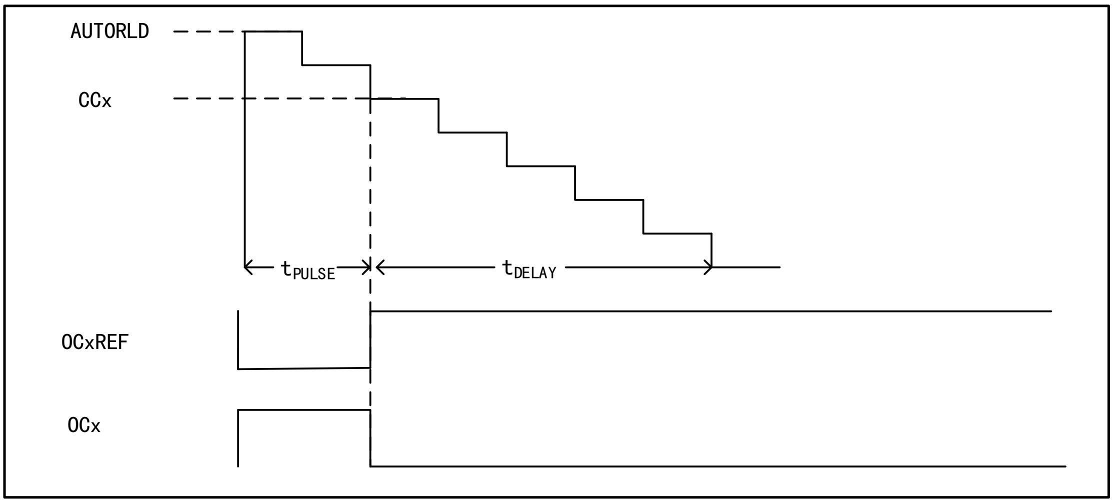
15.4.7 Single-pulse Mode

The single-pulse mode is a special case of timer comparison output, and is also a special case of PWM output mode.

Set SPMEN bit of TMRx_CTRL1 register, and select the single-pulse mode. After the counter is started, a certain number of pulses will be output before the update event occurs. When an update event occurs, the counter will stop counting, and the subsequent PWM waveform output will no longer be changed.

After a certain controllable delay, a pulse with controllable pulse width is generated in single-pulse mode through the program. The delay time is defined by the value of TMRx_CCx register; in the count-up mode, the delay time is CCx and the pulse width is AUTORLD-CCx; in the count-down mode, the delay time is AUTORLD-CCx and the pulse width is CCx.

Figure 36 Timing Diagram in Single-pulse Mode



15.4.8 Impact of the Register on Output Waveform

The following registers will affect the level of the timer output waveform. For details, please refer to "Register Functional Description".

（1） CCxEN and CCxNEN bits in TMRx_CCEN register

- CCxNEN=0 and CCxEN=0: The output is turned off (output disabled, invalid state)
- CCxNEN=1 and CCxEN=1: The output is turned on (output enabled, normal output)

（2） MOEN bit in TMRx_BDT register

- MOEN=0: Idle mode
- MOEN=1: Run mode

（3） OCxOIS and OCxNOIS bits in TMRx_CTRL2 register

- OCxOIS=0 amd OCxNOIS=0: When idle (MOEN=0), the output level after the dead-time is 0
- OCxOIS=1 amd OCxNOIS=1: When idle (MOEN=0), the output level after the dead-time is 1

（4） RMOS bit in TMRx_BDT register

- Application environment of RMOS: In corresponding complementary channel and timer are in run mode (MOEN=1), the timer is not working (CCxEN=0, CCxNEN=0) or is working (CCxEN=1, CCxNEN=1)

（5） IMOS bit in TMRx_BDT register

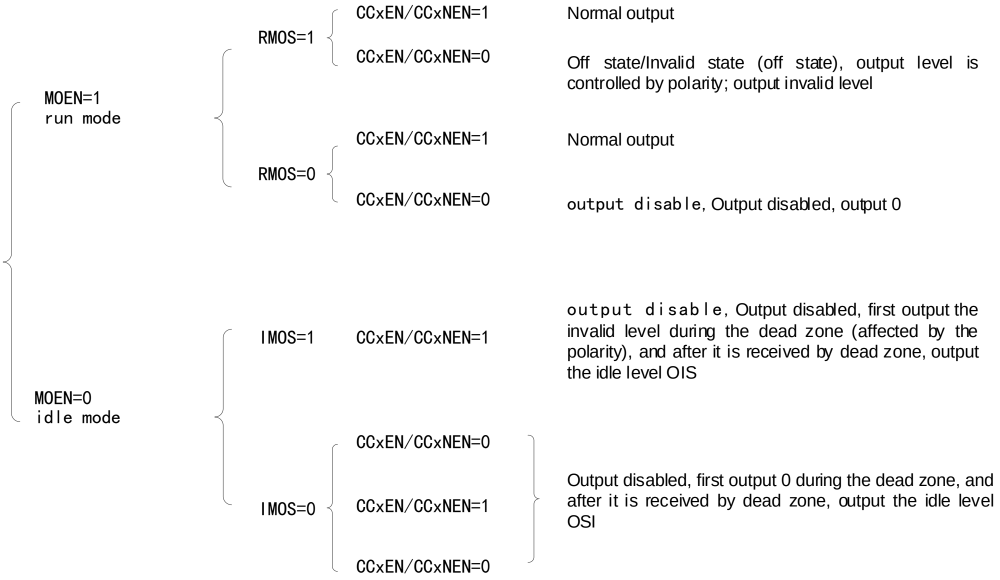
- Application environment of IMOS: In corresponding complementary channel and timer are in idle mode (MOEN=0), the timer is not working (CCxEN=0, CCxNEN=0) or is working (CCxEN=1, CCxNEN=1)

（6） CCxPOL and CCxNPOL bits of TMRx_CCEN register

- CCxPOL=0 and CCxNPOL=0: Output polarity, high level is valid
  CCxPOL=1 and CCxNPOL=1: Output polarity, the low level is valid

The following figure lists the register structure relationships that affect the output waveform

Figure 37 Register Structural Relationship Affecting Output Waveform


MOEN=1
run mode
RMOS=1
CCxEN/CCxNEN=1
Normal output
CCxEN/CCxNEN=0
Off state/Invalid state (off state), output level is controlled by polarity; output invalid level
RMOS=0
CCxEN/CCxNEN=1
Normal output
CCxEN/CCxNEN=0
output disable, Output disabled, output 0
MOEN=0
idle mode
IMOS=1
CCxEN/CCxNEN=1
output disable, Output disabled, first output the invalid level during the dead zone (affected by the polarity), and after it is received by dead zone, output the idle level OIS
IMOS=0
CCxEN/CCxNEN=0
CCxEN/CCxNEN=1
CCxEN/CCxNEN=0
Output disabled, first output 0 during the dead zone, and after it is received by dead zone, output the idle level OSI

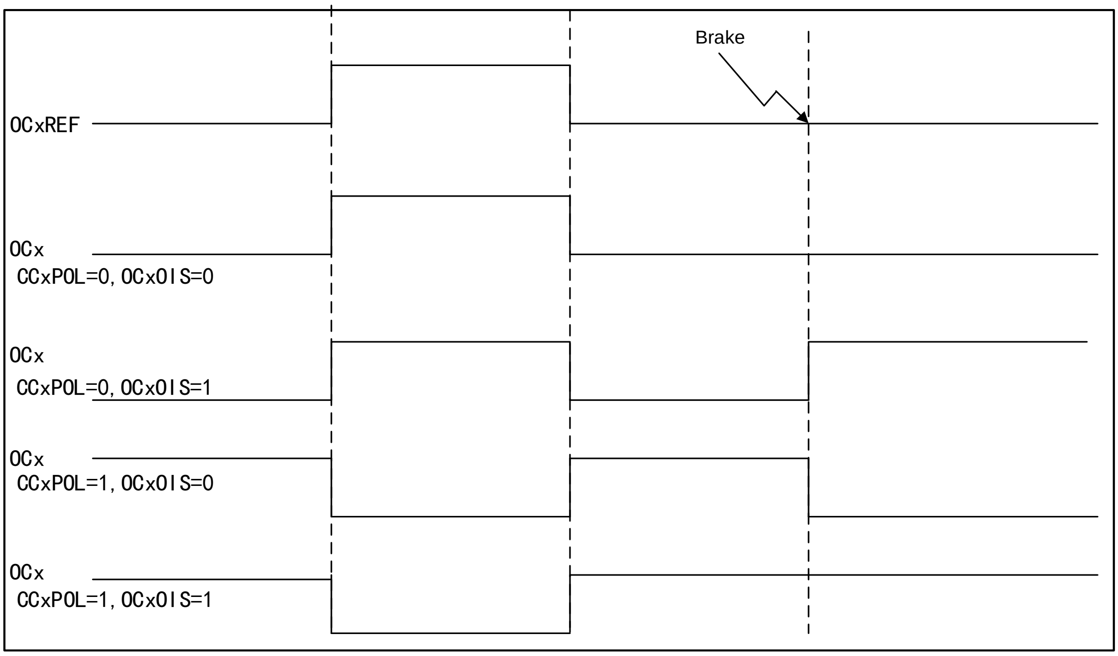
15.4.9 Braking Function

The signal source of braking is clock fault event and external input interface.

Besides, the BRKEN bit in TMRx_BDT register can enable the braking function, and the BRKPOL bit can configure the polarity of braking input signal.

When a braking event occurs, the output pulse signal level can be modified according to the state of the relevant control bit.

Figure 38 Braking Event Timing Diagram

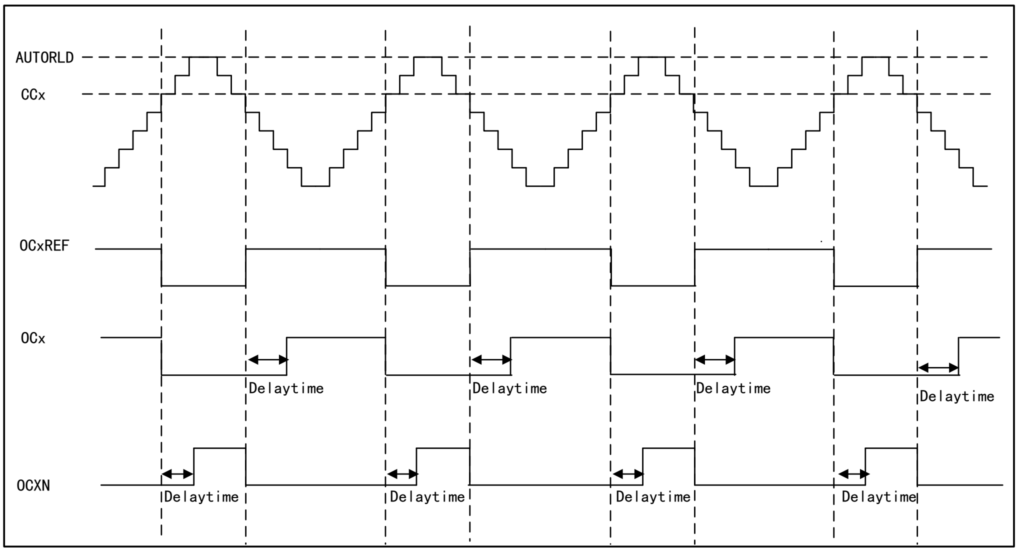


15.4.10 Complementary Output and Dead-time Insertion

Complementary output is particular output of advanced timer, and the advanced timer has three groups of complementary output channels. The insertion dead time is used to generate complementary output signals to ensure that the two-way complementary signals of channels will not be valid at the same time. The dead time is set according to the output device connected to the timer and its characteristics

The duration of the dead-time can be controlled by configuring DTS bit of TMRx_BDT register

Figure 39 Complementary Output of Insertion with Dead-time



15.4.11 Forced Output Mode

In the forced output mode, the comparison result is ignored, and the corresponding level is directly output according to the configuration instruction.

- CCxSEL=00 for TMRx_CCMx register, set CCx channel as output
- OCxMOD=100/101 for TMRx_CCMx register, set to force OCxREF signal to invalid/valid state

In this mode, the corresponding interrupt and DMA request will still be generated.

15.4.12 Encoder Interface Mode

The encoder interface mode is equivalent to an external clock with direction selection. In the encoder interface mode, the content of the timer can always indicate the position of the encoder.

The selection methods of encoder interface is as follows:

- By setting SMFSEL bit of TMRx_SMCTRL register, set the counter to count on the edge of TI1 channel /TI2 channel, or count on the edge of TI1 and TI2 at the same time.
- Select the polarity of TI1 and TI2 by setting the CC1POL and CC2POL bits of TMRx_CCEN register.
- Select to filter or not by setting the IC1F and IC2F bits of TMRx_CCM1 register.

The two input TI1 and TI2 can be used as the interface of incremental encoder. The counter is driven by the effective jump of the signals TI1FP1 and TI2FP2 after filtering and edge selection in TI1 and TI2.

The count pulse and direction signal are generated according to the input signals of TI1 and TI2

- The counter will count up/down according to the jumping sequence of the input signal.
- Set CNTDIR of control register TMRx_CTRL1 to be read-only (CNTDIR will be re-calculated due to jumping of any input end).

The change mechanism of counter count direction is shown in the figure below

Table 69 Relationship between Count Direction and Encoder

Effective edge		**Count only in TI1**		**Count only in TI2**		**Count in both TI1 and TI2**	
Level of relative signal		High	Low	High	Low	High	Low
TI1FP1	Rising edge	—		Count down	Count up	Count down	Count up
	Falling edge			Count up	Count down	Count up	Count down
TI2FP2	Rising edge	Count up	Count down	—		Count up	Count down

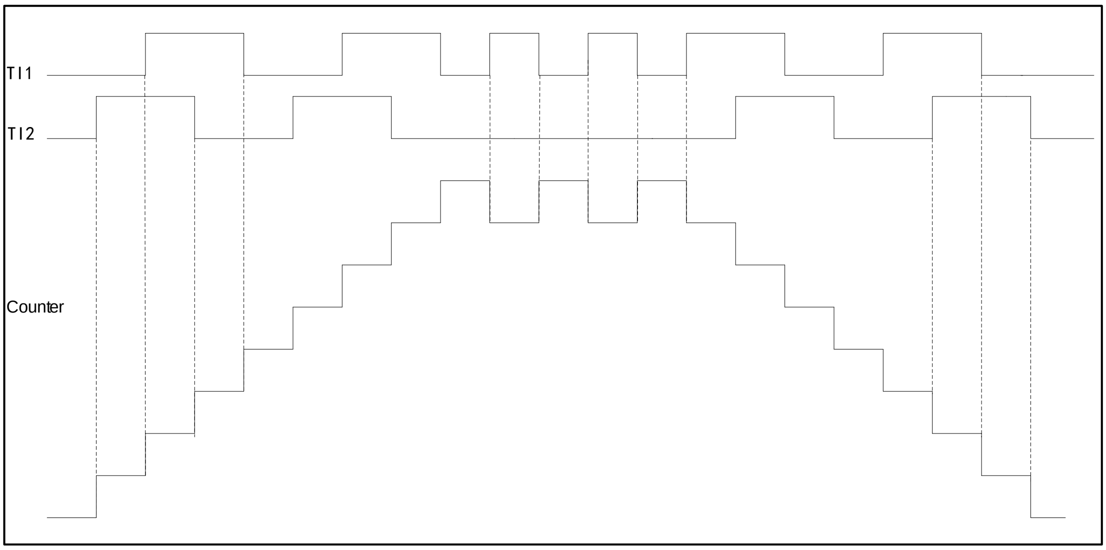
Effective edge		Count only in TI1		Count only in TI2	Count in both TI1 and TI2	
	Falling edge	Count down	Count up		Count down	Count up

The external incremental encoder can be directly connected with MCU, not needing external interface logic, so the comparator is used to convert the differential output of the encoder to digital signal to increase the immunity from noise interference.

Among the following examples:

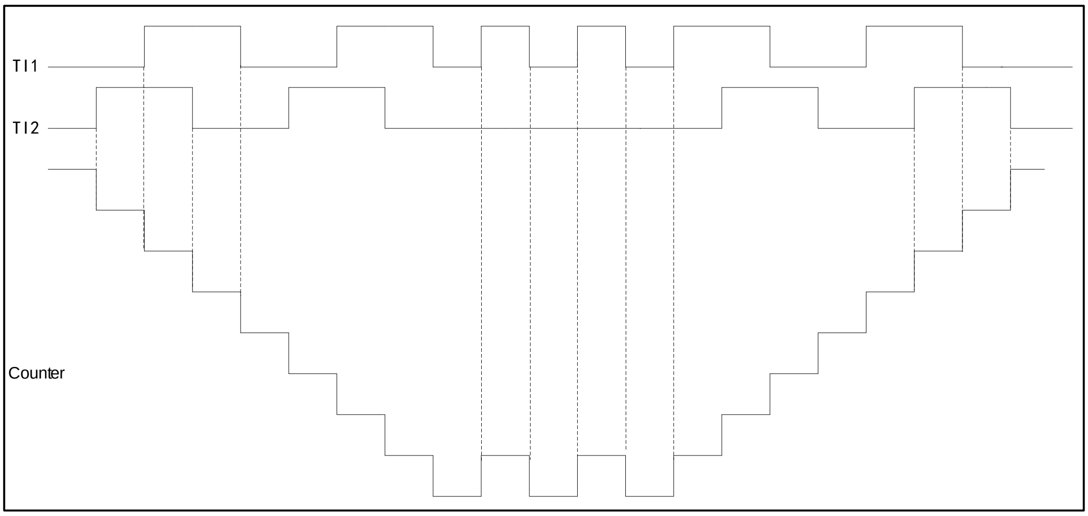
- IC1FP1 is mapped to TI1
- IC2FP2 is mapped to TI2
- Neither IC1FP1 nor IC2FP2 is reverse phase
- The input signal is valid at the rising edge and falling edge
- Enable the counter

Figure 40 Counter Operation Example in Encoder Mode



For example, when TI1 is at low level, and TI2 is in rising edge state, the counter will count up.

Figure 41 Example of Encoder Interface Mode of IC1FP1 Reversed Phase



For example, when TI1 is at low level, and the rising edge of TI2 jumps, the counter will count down.

15.4.13 Slave Mode

TMRx timer can synchronize external trigger

- Reset mode
- Gated mode
- Trigger mode

SMFSEL bit in TMRx_SMCTRL register can be set to select the mode.

SMFSEL=100 set the reset mode, SMFSEL=101 set the gated mode, SMFSEL=110 set the trigger mode.

In the reset mode, when a trigger input event occurs, the counter and prescaler will be initialized, and the rising edge of the selected trigger input (TRGI) will reinitialize the counter and generate a signal to update the register.

In the gated mode, the enable of the counter depends on the high level of the selected input. When the trigger input is high, the clock of the counter will be started. Once the trigger input becomes low, the counter will stop (but not be reset). The start and stop of the counter are controlled.

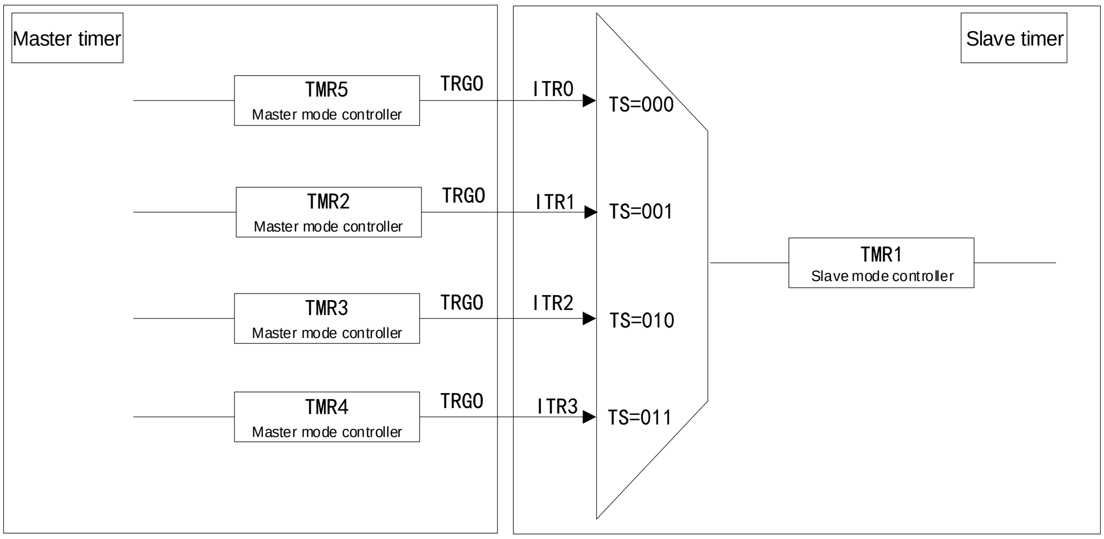
In the trigger mode, the enable of the counter depends on the event on the selected input, the counter is started (but is not reset) at the rising edge of the trigger input, and only the start of the counter is controlled.

15.4.14 Timer Interconnection

Each timer of TMRx can be connected with each other to realize synchronization or cascading between timers. It is required to configure one timer in master mode and the other timer in slave mode.

When the timer is in master mode, it can reset, start, stop and provide clock source for the counter of the slave mode timer.

Figure 42 Timer 1 Master/Slave Mode Example



When the timers are interconnected:

- A timer can be used as the prescaler of other register
- Another register can be started by the enable signal of a timer
- Another register can be started by the update event of a timer
- Another register can be selected by the enable of a timer
- Two timers can be synchronized by an external trigger

15.4.15 Interrupt and DMA Request

The timer can generate an interrupt when an event occurs during operation

- Update event (counter overrun/underrun, counter initialization)
- Trigger event (counter start, stop, internal/external trigger)
- Capture/Compare event
- Braking signal input event

Some internal interrupt events can generate DMA requests, and special interfaces can enable or disable DMA requests.

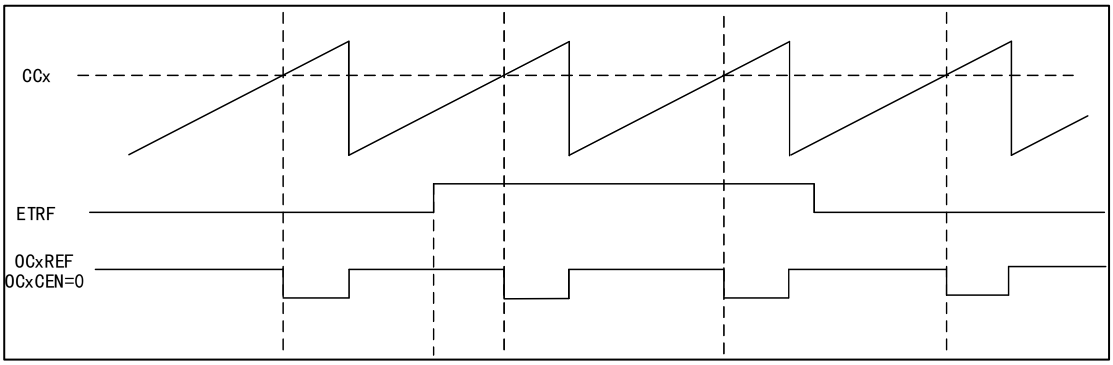
15.4.16 Clear OCxREF Signal when External Events Occur

This function is used for output compare and PWM mode.

In one channel, the high level of ETRF input port will reduce the signal of OCxREF to low level, and the OCxCEN bit in capture/compare register TMRx_CCMx is set to 1, and OCxREF signal will remain low until the next update event.

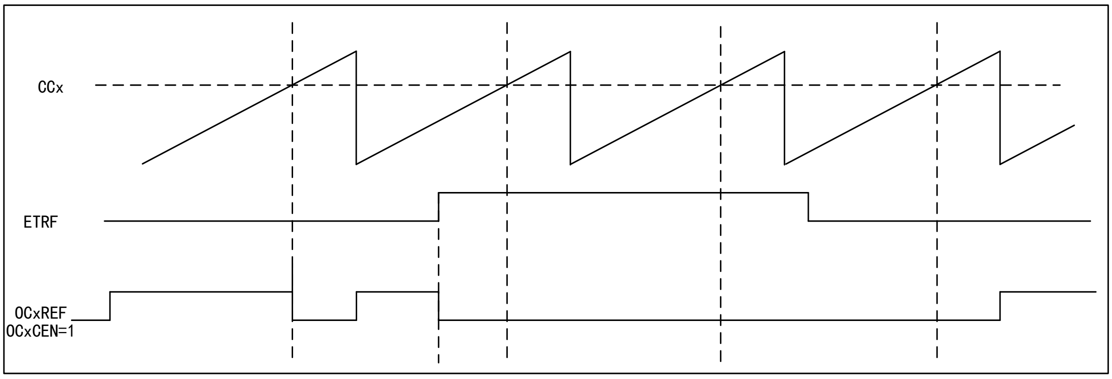
Set TMRx to PWM mode, close the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=0, and the output OCxREF signal is shown in the figure below.

Figure 43 OCxREF Timing Diagram



Set TMRx to PWM mode, close the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=1, and the output OCxREF signal is shown in the figure below.

Figure 44 OCxREF Timing Diagram



15.5 Register Address Mapping

In the following table, all registers of the advanced timer are mapped to a 16-bit addressable (address) space.

Table 70 Advanced Timer Register Address Mapping

Register name	Description	Offset address
TMRx_CTRL1	Control register 1	0x00
TMRx_CTRL2	Control register 2	0x04
TMRx_SMCTRL	Slave mode control register	0x08
TMRx_DIEN	DMA/Interrupt enable register	0x0C
TMRx_STS	State register	0x10
TMRx_CEG	Control event generation register	0x14
TMRx_CCM1	Capture/Compare mode register 1	0x18
TMRx_CCM2	Capture/Compare mode register 2	0x1C
TMRx_CCEN	Capture/Compare enable register	0x20

Register name	Description	Offset address
TMRx_CNT	Counter register	0x24
TMRx_PSC	Prescaler register	0x28
TMRx_AUTORLD	Auto reload register	0x2C
TMRx_REPCNT	Repeat count register	0x30
TMRx_CC1	Channel 1 capture/compare register	0x34
TMRx_CC2	Channel 2 capture/compare register	0x38
TMRx_CC3	Channel 3 capture/compare register	0x3C
TMRx_CC4	Channel 4 capture/compare register	0x40
TMRx_BDT	Brake and dead-time register	0x44
TMRx_DCTRL	DMA control register	0x48
TMRx_DMADDR	DMA address register of continuous mode	0x4C

15.6 Register Functional Description

15.6.1 Control register 1 (TMRx_CTRL1)

Offset address: 0x00

Reset value: 0x0000

Field	Name	R/W	Description
0	CNTEN	R/W	Counter Enable 0: Disable 1: Enable When the timer is configured as external clock, gated mode and encoder mode, it is required to write 1 to the bit by software to start regular work; when it is configured as the trigger mode, it can be written to 1 by hardware.
1	UD	R/W	Update Disable Update event can cause AUTORLD, PSC and CCx to generate the value of update setting. 0: Update event is allowed (UEV) An update event can occur in any of the following situations: The counter overruns/underruns; Set UEG bit; Update generated by slave mode controller. 1: Update event is disabled
2	URSSEL	R/W	Update Request Source Select If interrupt or DMA is enabled, the update event can generate update interrupt or DMA request. Different update request sources can be selected through this bit. 0: The counter overruns or underruns Set UEG bit; Update generated by slave mode controller.

Field	Name	R/W	Description
			1: The counter overruns or underruns
3	SPMEN	R/W	Single Pulse Mode Enable When an update event is generated, the output level of the channel can be changed; in this mode, the CNTEN bit will be cleared, the counter will be stopped, and the output level of the channel will not be changed. 0: Disable 1: Enable
4	CNTDIR	R/W	Counter Direction This bit is read-only when the counter is configured as center-aligned mode or encoder mode. 0: Count up 1: Count down
6:5	CAMSEL	R/W	Center Aligned Mode Select In the center-aligned mode, the counter counts up and down alternately; otherwise, it will only count up or down. Different center-aligned modes affect the timing of setting the output compare interrupt flag bit of the output channel to 1; when the counter is disabled (CNTEN=0), select the center-aligned mode. 00: Edge alignment mode 01: Center-aligned mode 1 (the output compare interrupt flag bit of output channel is set to 1 when counting down) 10: Center-aligned mode 2 (the output compare interrupt flag bit of output channel is set to 1 when counting up) 11: Center-aligned mode 3 (the output compare interrupt flag bit of output channel is set to 1 when counting up/down)
7	ARPEN	R/W	Auto-reload Preload Enable When the buffer is disabled, the program modification TMRx_AUTORLD will immediately modify the values loaded to the counter; when the buffer is enabled, the program modification TMRx_AUTORLD will modify the values loaded to the counter in the next update event. 0: Disable 1: Enable
9:8	CLKDIV	R/W	Clock Divide Factor For the configuration of dead time and digital filter, CK_INT provides the clock, and the dead time and the clock of the digital filter can be adjusted by setting this bit. 00: $t_{DTS}=t_{CK\_INT}$ 01: $t_{DTS}=2\times t_{CK\_INT}$ 10: $t_{DTS}=4\times t_{CK\_INT}$ 11: Reserved
15:10	Reserved		

15.6.2 Control register 2 (TMRx_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
0	CCPEN	R/W	Capture/Compare Preloaded Enable This bit affects the change of CCxEN, CCxNEN and OCxMOD values. When preloading is disabled, the program modification will immediately affect the timer setting; When preloading is enabled, it is only updated after COMG is set, so as to affect the setting of timer; this bit only works on channels with complementary output. 0: Disable 1: Enable
1	Reserved		
2	CCUSEL	R/W	Capture/compare Control Update Select Only when the capture/compare preload is enabled (CCPEN=1), it works only for complementary output channel. 0: It can only be updated by setting COMG bit 1: It can be updated by setting COMG bit or rising edge on TRGI
3	CCDSEL	R/W	Capture/compare DMA Select 0: Send DMA request of CCx when CCx event occurs 1: Send DMA request of CCx when an update event occurs
6:4	MMSEL	R/W	Master Mode Signal Select The signals of timers working in master mode can be used for TRGO, which affects the work of timers in slave mode and cascaded with master timer, and specifically affects the configuration of timers in slave mode. 000: Reset; the reset signal of master mode timer is used for TRGO 001: Enable; the counter enable signal of master mode timer is used for TRGO 010: Update; the update event of master mode timer is used for TRGO 011: Compare pulses; when the master mode timer captures/compares successfully (CCxIFLG=1), a pulse signal is output for TRGO 100: Compare mode 1; OC1REF is used to trigger TRGO 101: Compare mode 2; OC2REF is used to trigger TRGO 110: Compare mode 3; OC3REF is used to trigger TRGO 111: Compare mode 4; OC4REF is used to trigger TRGO
7	TI1SEL	R/W	Timer Input 1 Selection 0: TMRx_CH1 pin is connected to TI1 input 1: TMRx_CH1, TMRx_CH2 and TMRx_CH3 pins are connected to TI1 input after exclusive
8	OC1OIS	R/W	OC1 Output Idle State Configure Only the level state after the dead time of OC1 is affected when MOEN=0 and OC1N is realized. 0：OC1=0 1：OC1=1 Note: When LOCKCFG bit in TMRx_BDT register is at the Level 1, 2 or 3, this bit cannot be modified.

Field	Name	R/W	Description
9	OC1NOIS	R/W	OC1N Output Idle State Configure Only the level state after the dead time of OC1 is affected when MOEN=0 and OC1N is realized. 0：OC1N=0 1：OC1N=1 Note: When PLOCKCFG bit in TMRx_BDT register is at the Level 1, 2 or 3, this bit cannot be modified.
10	OC2OIS	R/W	Configure OC2 output idle state. Refer to OC1OIS bit
11	OC2NOIS	R/W	Configure OC2N output idle state. Refer to OC1NOIS bit
12	OC3OIS	R/W	Configure OC3 output idle state. Refer to OC1OIS bit
13	OC3NOIS	R/W	Configure OC3N output idle state. Refer to OC1NOIS bit
14	OC4OIS	R/W	Configure OC4 output idle state. Refer to OC1OIS bit
15	Reserved		

15.6.3 Slave mode control register (TMRx_SMCTRL)

Offset address: 0x08

Reset value: 0x0000

Field	Name	R/W	Description
2:0	SMFSEL	R/W	Slave Mode Function Select 000: Disable the slave mode, the timer can be used as master mode timer to affect the work of slave mode timer; if CTRL1_CNTEN=1, the prescaler is directly driven by the internal clock. 001: Encoder mode 1; according to the level of TI1FP1, the counter counts at the edge of TI2FP2. 010: Encoder mode 2; according to the level of TI2FP2, the counter counts at the edge of TI1FP1. 011: Encoder mode 3; according to the input level of another signal, the counter counts at the edge of TI1FP1 and TI2FP2. 100: Reset mode; the slave mode timer resets the counter after receiving the rising edge signal of TRGI and generates the signal to update the register. 101: Gated mode; the slave mode timer starts the counter to work after receiving the TRGI high level signal; it stops the counter when receiving TRGI low level; when receiving TRGI high level signal again, the timer will continue to work; the counter is not reset during the whole period. 110: Trigger mode, the slave mode timer starts the counter to work after receiving the rising edge signal of TRGI. 111: External clock mode 1; select the rising edge signal of TRGI as the clock source to drive the counter to work.
3	Reserved		
6:4	TRGSEL	R/W	Trigger Input Signal Select In order to avoid false edge detection when changing the bit value, it must be changed when SMFSEL=0. 000: Internal trigger ITR0 001: Internal trigger ITR1

Field	Name	R/W	Description
			010: Internal trigger ITR2 011: Internal trigger ITR3 100: Channel 1 input edge detector TIF_ED 101: Channel 1 post-filtering timer input TI1FP1 110: Channel 2 post-filtering timer input TI2FP2 111: External trigger input (ETRF)
7	MSMEN	R/W	Master/slave Mode Enable 0: Invalid 1: Enable the master/slave mode
11:8	ETFCFG	R/W	External Trigger Filter Configure 0000: Filter disabled, sampling by f_{DTS} 0001：DIV=1, N=2 0010：DIV=1, N=4 0011：DIV=1, N=8 0100：DIV=2, N=6 0101：DIV=2, N=8 0110：DIV=4, N=6 0111：DIV=4, N=8 1000：DIV=8, N=6 1001：DIV=8, N=8 1010：DIV=16, N=5 1011：DIV=16, N=6 1100：DIV=16, N=8 1101：DIV=32, N=5 1110：DIV=32, N=6 1111：DIV=32, N=8 Sampling frequency=timer clock frequency/DIV; the filter length=N, and a jump is generated by every N events.
13:12	ETPCFG	R/W	External Trigger Prescaler Configure The ETR (external trigger input) signal becomes ETRP after frequency division. The signal frequency of ETRP is at most 1/4 of TMRxCLK frequency; when ETR frequency is too high, the ETRP frequency must be reduced through frequency division. 00: The prescaler is disabled; 01: ETR signal 2 divided frequency 10: ETR signal 4 divided frequency 11: ETR signal 8 divided frequency
14	ECEN	R/W	External Clock Mode2 Enable 0: Disable 1: Enable Setting ECEN bit has the same function as selecting external clock mode 1 to connect TRGI to ETRF; slave mode (reset, gating, trigger) can be used at the same time with external clock mode 2, but TRGI cannot be connected to ETRF in such case; when external clock mode 1 and external clock mode 2 are enabled at the same time, the input of external clock is ETRF.
15	ETPOL	R/W	External Trigger Polarity Configure This bit decides whether the external trigger ETR is reversed.

Field	Name	R/W	Description
			0: The external trigger ETR is not reversed,and the high level or rising edge is valid 1: The external trigger ETR is reversed, and the low level or falling edge is valid

Table 71 TMRx Internal Trigger Connection

Slave timer	ITR1 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TMR1	TMR5	TMR2	TMR3	TMR4
TMR8	TMR1	TMR2	TMR4	TMR5

15.6.4 DMA/Interrupt enable register (TMRx_DIEN)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
0	UIEN	R/W	Update interrupt Enable 0: Disable 1: Enable
1	CC1IEN	R/W	Capture/Compare Channel1 Interrupt Enable 0: Disable 1: Enable
2	CC2IEN	R/W	Capture/Compare Channel2 Interrupt Enable 0: Disable 1: Enable
3	CC3IEN	R/W	Capture/Compare Channel3 Interrupt Enable 0: Disable 1: Enable
4	CC4IEN	R/W	Capture/Compare Channel4 Interrupt Enable 0: Disable 1: Enable
5	COMIEN	R/W	COM Interrupt Enable 0: Disable 1: Enable
6	TRGIEN	R/W	Trigger interrupt Enable 0: Disable 1: Enable
7	BRKIEN	R/W	Break interrupt Enable 0: Disable 1: Enable
8	UDIEN	R/W	Update DMA Request Enable 0: Disable 1: Enable
9	CC1DEN	R/W	Capture/Compare Channel1 DMA Request Enable 0: Disable 1: Enable

Field	Name	R/W	Description
10	CC2DEN	R/W	Capture/Compare Channel2 DMA Request Enable 0: Disable 1: Enable
11	CC3DEN	R/W	Capture/Compare Channel3 DMA Request Enable 0: Disable 1: Enable
12	CC4DEN	R/W	Capture/Compare Channel4 DMA Request Enable 0: Disable 1: Enable
13	COMDEN	R/W	COM DMA Request Enable 0: Disable 1: Enable
14	TRGDEN	R/W	Trigger DMA Request Enable 0: Disable 1: Enable
15	Reserved		

15.6.5 State register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
0	UIFLG	RC_W0	Update Event Interrupt Generate Flag 0: Update event interrupt does not occur 1: Update event interrupt occurs When the counter value is reloaded or reinitialized, an update event will be generated. The bit is set to 1 by hardware and cleared by software; update events are generated in the following situations: (1) UD=0 on TMRx_CTRL1 register, and when the value of the repeat counter overruns/underruns, an update event will be generated; (2) URSSEL=0 and UD=0 on TMRx_CTRL1 register, configure UEG=1 on TMRx_CEG register to generate update event, and the counter needs to be initialized by software; (3) URSSEL=0 and UD=0 on TMRx_CTRL1 register, generate update event when the counter is initialized by trigger event.

Field	Name	R/W	Description
1	CC1IFLG	RC_W0	Capture/Compare Channel1 Interrupt Flag **When the capture/compare channel 1 is configured as output:** 0: No matching occurred 1: The value of TMRx_CNT matches the value of TMRx_CC1 **When the capture/compare channel 1 is configured as input:** 0: Input capture did not occur 1: Input capture occurred When capture event occurs, the bit is set to 1 by hardware, and it can be cleared by software or cleared when reading TMRx_CC1 register.
2	CC2IFLG	RC_W0	Captuer/Compare Channel2 Interrupt Flag Refer to STS_CC1IFLG
3	CC3IFLG	RC_W0	Capture/Compare Channel3 Interrupt Flag Refer to STS_CC1IFLG
4	CC4IFLG	RC_W0	Captuer/Compare Channel4 Interrupt Flag Refer to STS_CC1IFLG
5	COMIFLG	RC_W0	COM Event Interrupt Generate Flag 0: COM event does not occur 1: COM interrupt waits for response After COM event is generated, this bit is set to 1 by hardware and cleared by software.
6	TRGIFLG	RC_W0	Trigger Event Interrupt Generate Flag 0: Trigger event interrupt did not occur 1: Trigger event interrupt occurred After Trigger event is generated, this bit is set to 1 by hardware and cleared by software.
7	BRKIFLG	RC_W0	Brake Event Interrupt Generate Flag 0: Brake event does not occur 1: Brake event occurs When brake input is valid, this bit is set to 1 by hardware; when brake input is invalid, this bit can be cleared by software.
8	Reserved		
9	CC1RCFLG	RC_W0	Capture/compare Channel1 Repetition Capture Flag 0: Repeat capture does not occur 1: Repeat capture occurs The value of the counter is captured to TMRx_CC1 register, and CC1IFLG=1; this bit is set to 1 by hardware and cleared by software only when the channel is configured as input capture.
10	CC2RCFLG	RC_W0	Capture/compare Channel2 Repetition Capture Flag Refer to STS_CC1RCFLG
11	CC3RCFLG	RC_W0	Capture/compare Channel3 Repetition Capture Flag Refer to STS_CC1RCFLG

Field	Name	R/W	Description
12	CC4RCFLG	RC_W0	Capture/compare Channel4 Repetition Capture Flag Refer to STS_CC1RCFLG
15:13	Reserved		

15.6.6 Control event generation register (TMRx_CEG)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
0	UEG	W	Update Event Generate 0: Invalid 1: Initialize the counter and generate the update event This bit is set to 1 by software, and cleared by hardware. Note: When an update event is generated, the counter of the prescaler will be cleared, but the prescaler factor remains unchanged. In the count-down mode, the counter reads the value of TMRx_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared.
1	CC1EG	W	Capture/Compare Channel1 Event Generation 0: Invalid 1: Capture/Compare event is generated This bit is set to 1 by software and cleared automatically by hardware. If Channel 1 is in output mode, When CC1IFLG=1, if CC1IEN and CC1DEN bits are set, the corresponding interrupt and DMA request will be generated. If Channel 1 is in input mode The value of the capture counter is stored in TMRx_CC1 register; configure CC1IFLG=1, and if CC1IEN and CC1DEN bits are also set, the corresponding interrupt and DMA request will be generated; at this time, if CC1IFLG=1, it is required to configure CC1RCFLG=1.
2	CC2EG	W	Capture/Compare Channel2 Event Generation Refer to CC1EG description
3	CC3EG	W	Capture/Compare Channel3 Event Generation Refer to CC1EG description
4	CC4EG	W	Capture/Compare Channel4 Event Generation Refer to CC1EG description
5	COMG	W	Capture/Compare Control Update Event Generate 0: Invalid 1: Capture/Compare update event is generated This bit is set to 1 by software and cleared automatically by hardware. Note: COMG bit is valid only in complementary output channel.
6	TEG	W	Trigger Event Generate 0: Invalid 1: Trigger event is generated This bit is set to 1 by software and cleared automatically by hardware.

Field	Name	R/W	Description
7	BEG	W	Brake Event Generate 0: Invalid 1: Brake event is generated This bit is set to 1 by software and cleared automatically by hardware.
15:8	Reserved		

15.6.7 Capature/Compare mode register 1 (TMRx_CCM1)

Offset address: 0x18

Reset value: 0x0000

The timer can be configured as input (capture mode) or output (compare mode) by CCxSEL bit. The functions of other bits of the register are different in input and output modes, and the functions of the same bit are different in output mode and input mode. The OCX in the register describes the function of the channel in the output mode, and the ICx in the register describes the function of the channel in the input mode.

Output compare mode:

Field	Name	R/W	Description
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Selection This bit defines the input/output direction and the selected input pin. 00: CC1 channel is output 01: CC1 channel is input, and IC1 is mapped on TI1 10: CC1 channel is input, and IC1 is mapped on TI2 11: CC1 channel is input, and IC1 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC1EN=0).
2	OC1FEN	R/W	Output Compare Channel1 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/compare output to the trigger input event.
3	OC1PEN	R/W	Output Compare Channel1 Preload Enable 0: Preloading function is disabled; write the value of TMRx_CC1 register through the program and it will work immediately. 1: Preloading function is enabled; write the value of TMRx_CC1 register through the program and it will work after an update event is generated. Note: When the protection level is 3 and the channel is configured as output, this bit cannot be modified. When the preload register is uncertain, PWM mode can be used only in single pulse mode (SPMEN=1); otherwise, the following output compare result is uncertain.
6:4	OC1MOD	R/W	Output Compare Channel1 Mode Configure 000: Freeze. The output compare has no effect on OC1REF 001: The output value is high when matching. When the value of counter CNT matches the value CCx of capture comparison register, OC1REF will be forced to be at high level

Field	Name	R/W	Description
			010: The output value is low when matching. When the value of the counter matches the value of the capture comparison register, OC1REF will be forced to be at low level 011: Output flaps when matching. When the value of the counter matches the value of the capture comparison register, flap the level of OC1REF 100: The output is forced to be low. Force OC1REF to be at low level 101: The output is forced to be high. Force OC1REF to be at high level 110: PWM mode 1 (set to high when the counter value<output compare value; otherwise, set to low) 111: PWM mode 2 (set to high when the counter value>output compare value; otherwise, set to low) Note: When the protection level is 3 and the channel is configured as output, this bit cannot be modified. In PWM modes 1 and 2, the OC1REF level changes when the comparison result changes or when the output compare mode changes from freeze mode to PWM mode.
7	OC1CEN	R/W	Output Compare Channel1 Clear Enable 0: OC1REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0
9:8	CC2SEL	R/W	Capture/Compare Channel2 Select This bit defines the input/output direction and the selected input pin. 00: CC2 channel is output 01: CC2 channel is input, and IC2 is mapped on TI2 10: CC2 channel is input, and IC2 is mapped on TI1 11: CC2 channel is input, and IC2 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC2EN=0).
10	OC2FEN	R/W	Output Compare Channel2 Preload Enable
11	OC2PEN	R/W	Output Compare Channel2 Buffer Enable
14:12	OC2MOD	R/W	Output Compare Channel1 Mode
15	OC2CEN	R/W	Output Compare Channel2 Clear Enable

Input capture mode:

Field	Name	R/W	Description
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Select 00: CC1 channel is output 01: CC1 channel is input, and IC1 is mapped on TI1 10: CC1 channel is input, and IC1 is mapped on TI2 11: CC1 channel is input, and IC1 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN bit CC1EN=0).
3:2	IC1PSC	R/W	Input Capture Channel 1 Perscaler Configure 00：PSC=1 01：PSC=2 10：PSC=4

Field	Name	R/W	Description
			11：PSC=8 PSC is prescaled factor, which triggers capture once every PSC events.
7:4	IC1F	R/W	Input Capture Channel 1 Filter Configuration 0000: Filter disabled, sampling by f_{DTS} 0001：DIV=1, N=2 0010：DIV=1, N=4 0011：DIV=1, N=8 0100：DIV=2, N=6 0101：DIV=2, N=8 0110：DIV=4, N=6 0111：DIV=4, N=8 1000：DIV=8, N=6 1001：DIV=8, N=8 1010：DIV=16, N=5 1011：DIV=16, N=6 1100：DIV=16, N=8 1101：DIV=32, N=5 1110：DIV=32, N=6 1111：DIV=32, N=8 Sampling frequency=timer clock frequency/DIV; the filter length=N, indicating that a jump is generated by every N events.
9:8	CC2SEL	R/W	Capture/Compare Channel 2 Select 00: CC2 channel is output 01: CC2 channel is input, and IC2 is mapped on TI1 10: CC2 channel is input, and IC2 is mapped on TI2 11: CC2 channel is input, and IC2 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC2EN=0).
11:10	IC2PSC	R/W	Input Capture Channel 2 Perscaler Configuration
15:12	IC2F	R/W	Input Capture Channel 2 Filter Configuration

15.6.8 Capture/Comparison mode register 2 (TMRx_CCM2)

Offset address: 0x1C

Reset value: 0x0000

Refer to the description of the above CCM1 register.

Output compare mode:

Field	Name	R/W	Description
1:0	CC3SEL	R/W	Capture/Compare Channel 1 Selection This bit defines the input/output direction and the selected input pin. 00: CC3 channel is output 01: CC3 channel is input, and IC3 is mapped on TI3 10: CC3 channel is input, and IC3 is mapped on TI4 11: CC3 channel is input, and IC3 is mapped on TRC, and only works in internal trigger input

Field	Name	R/W	Description
			Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC3EN=0).
2	OC3FEN	R/W	Output Compare Channel3 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/comparison output to the trigger input event.
3	OC3PEN	R/W	Output Compare Channel3 Preload Enable
6:4	OC3MOD	R/W	Output Compare Channel3 Mode Configure
7	OC3CEN	R/W	Output Compare Channel3 Clear Enable 0: OC3REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Selection This bit defines the input/output direction and the selected input pin. 00: CC4 channel is output 01: CC4 channel is input, and IC4 is mapped on TI4 10: CC4 channel is input, and IC4 is mapped on TI3 11: CC4 channel is input, and IC4 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC4EN=0).
10	OC4FEN	R/W	Output Compare Channel4 Preload Enable
11	OC4PEN	R/W	Output Compare Channel4 Buffer Enable
14:12	OC4MOD	R/W	Output Compare Channel4 Mode Configure
15	OC4CEN	R/W	Output Compare Channel4 Clear Enable

Input capture mode:

Field	Name	R/W	Description
1:0	CC3SEL	R/W	Capture/Compare Channel 3 Select 00: CC3 channel is output 01: CC3 channel is input, and IC3 is mapped on TI3 10: CC3 channel is input, and IC3 is mapped on TI4 11: CC3 channel is input, and IC3 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC3EN=0).
3:2	IC3PSC	R/W	Input Capture Channel 3 Perscaler Configuration 00：PSC=1 01：PSC=2 10：PSC=4 11：PSC=8 PSC is prescaled factor, which triggers capture once every PSC events.
7:4	IC3F	R/W	Input Capture Channel 3 Filter Configuration
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Select 00: CC4 channel is output

Field	Name	R/W	Description
			01: CC4 channel is input, and IC4 is mapped on TI4 10: CC4 channel is input, and IC4 is mapped on TI3 11: CC4 channel is input, and IC4 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC4EN=0).
11:10	IC4PSC	R/W	Input Capture Channel 4 Perscaler Configuration
15:12	IC4F	R/W	Input Capture Channel 4 Filter Configuration

15.6.9 Capture/Comparison enable register (TMRx_CCEN)

Offset address: 0x20

Reset value: 0x0000

Field	Name	R/W	Description
0	CC1EN	R/W	Capture/Compare Channel1 Output Enable **When the capture/comparison channel 1 is configured as output:** 0: Output is disabled 1: Output is enabled **When the capture/comparison channel 1 is configured as input:** This bit determines whether the value CNT of the counter can be captured and enter TMRx_CC1 register 0: Capture is disabled 1: Capture is enabled
1	CC1POL	R/W	Capture/Compare Channel1 Output Polarity Configure **When CC1 channel is configured as output:** 0: OC1 high level is valid 1: OC1 low level is valid **When CC1 channel is configured as input:** 0: Phase not reversed: capture at the rising edge of IC1; phase not reversed when IC1 is used as external trigger. 1: Phase reversed, capature at the falling edge of ICC1; phase reversed when IC1 is used as external trigger. Note: When the protection level is 2 or 3, this bit cannot be modified
2	CC1NEN	R/W	Capture/Compare Channel1 Complementary Output Enable 0: Disable 1: Enable
3	CC1NPOL	R/W	Capture/Compare Channel1 Complementary Output Polarity 0: OC1N high level is valid 1: OC1N low level is valid Note: When the protection level is 2 or 3, this bit cannot be modified
4	CC2EN	R/W	Capture/Compare Channel2 Output Enable Refer to CCEN_CC1EN
5	CC2POL	R/W	Capture/Compare Channel2 Output Polarity Configure Refer to CCEN_CC1POL
6	CC2NEN	R/W	Capture/Compare Channel1 Complementary Output Enable Refer to CCEN_CC1NEN

Field	Name	R/W	Description
7	CC2NPOL	R/W	Capture/Compare Channel2 Complementary Output Polarity Configure Refer to CCEN_CC1NPOL
8	CC3EN	R/W	Capture/Compare Channel3 Output Enable Refer to CCEN_CC1EN
9	CC3POL	R/W	Capture/Compare Channel3 Output Polarity Configure Refer to CCEN_CC1POL
10	CC3NEN	R/W	Capture/Compare Channel3 Complementary Output Enable Refer to CCEN_CC1NEN
11	CC3NPOL	R/W	Capture/Compare Channel3 Complementary Output Polarity Configure Refer to CCEN_CC1NPOL
12	CC4EN	R/W	Capture/Compare Channel4 Output Enable Refer to CCEN_CC1EN
13	CC4POL	R/W	Capture/Compare Channel4 Output Polarity Refer to CCEN_CC1POL
15:14	Reserved		

15.6.10 Counter register (TMRx_CNT)

Offset address: 0x24

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CNT	R/W	Counter Value

15.6.11 Prescaler register (TMRx_PSC)

Offset address: 0x28

Reset value: 0x0000

Field	Name	R/W	Description
15:0	PSC	R/W	Prescaler Value Clock frequency of counter (CK_CNT)=f_{CK_PSC}/(PSC+1)

15.6.12 Auto reload register (TMRx_AUTORLD)

Offset address: 0x2C

Reset value: 0xFFFF

Field	Name	R/W	Description
15:0	AUTORLD	R/W	Auto Reload Value When the value of auto reload is empty, the counter will not count.

15.6.13 Repeat count register (TMRx_REPCNT)

Offset address: 0x30

Reset value: 0x0000

Field	Name	R/W	Description
7:0	REPCNT	R/W	Repetition Counter Value When the count value of the repeat counter is reduced to 0, an update event will be generated, and the counter will start counting again from the REPCNT value; the new value newly written to this register is valid only when an update event occurs in next cycle.
15:8	Reserved		

15.6.14 Channel 1 capture/comparison register (TMRx_CC1)

Offset address: 0x34

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC1	R/W	Capture/Compare Channel 1 Value **When the capture/comparison channel 1 is configured as input mode:** CC1 contains the counter value transmitted by the last input capture channel 1 event. **When the capture/comparison channel 1 is configured as output mode:** CC1 contains the current load capture/comparison register value Compare the value CC1 of the capture and comparison channel 1 with the value CNT of the counter to generate the output signal on OC1. When the output compare preload is disabled (OC1PEN=0 for TMRx_CCM1 register), the written value will immediately affect the output compare results; If the output compare preload is enabled (OC1PEN=1 for TMRx_CCM1 register), the written value will affect the output compare result when an update event is generated.

15.6.15 Channel 2 capture/comparison register (TMRx_CC2)

Offset address: 0x38

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC2	R/W	Capture/Compare Channel 2 Value Refer to TMRx_CC1

15.6.16 Channel 3 capture/comparison register (TMRx_CC3)

Offset address: 0x3C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC3	R/W	Capture/Compare Channel 3 Value Refer to TMRx_CC1

15.6.17 Channel 4 capture/comparison register (TMRx_CC4)

Offset address: 0x40

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC4	R/W	Capture/Compare Channel 4 Value Refer to TMRx_CC1

15.6.18 Brake and dead-time register (TMRx_BDT)

Offset address: 0x44

Reset value: 0x0000

Note: According to the lock setting, AOEN, BRKPOL, BRKEN, IMOS, RMOS and DTS[7:0] bits all can be write-protected, and it is necessary to configure them when writing to TMRx_BDT register for the first time.

Field	Name	R/W	Description
7:0	DTS	R/W	Dead Time Setup DT is the dead duration, and the relationship between DT and register DTS is as follows: DTS[7:5]=0xx=>DT=DTS[7:0]×T_{DTS}, T_{DTS}=TDTS； DTS[7:5]=10x=>DT=（64+DTS[5:0]）×T_{DTS}, T_{DTS}=2×T_{DTS}； DTS[7:5]=110=>DT=（32+DTS[4:0]）×T_{DTS}, T_{DTS}=8×T_{DTS}； DTS[7:5]=111=>DT=（32+DTS[4:0]）×T_{DTS}, T_{DTS}=16×T_{DTS}； For example: assuming T_{DTS}=125ns (8MHZ), the dead time setting is as follows: If the step time is 125ns, the dead time can be set from 0 to 15875ns; If the step time is 250ns, the dead time can be set from 16us to 31750ns; If the step time is 1us, the dead time can be set from 32us to 63us; If the step time is 2us, the dead time can be set from 64us to 126us. Note: Once LOCK level (LOCKCFG bit in TMRx_BDT register) is set to 1, 2 or 3, these bits cannot be modified.
9:8	LOCKCFG	R/W	Lock Write Protection Mode Configuration 00: Without Lock write protection level; the register can be written directly 01: Lock write protection level 1 It cannot be written to DTS, BRKEN, BRKPOL and AOEN bits of TMRx_BDT, and OCxOIS and OCxNOIS bits of TMRx_CTRL2 register. 02: Lock write protection level 2 It is not allowed to write to all bits with protection level 1 and write to the CCxPOL and OCxNPOL bits in TMRx_CCEN register and the RMOS and IMOS bits in TMRx_BDT register. 11: Lock write protection level 3 It is not allowed to write to all bits with protection level 2, and write to the OCxMOD and OCxPEN bits of TMRx_CCMx register. Note: After system reset, the lock write protect bit can only be written once.
10	IMOS	R/W	Idle Mode Off-state Configure Idle mode means MOEN=0; disabled means CcxEN=0; this bit describes the impact of different values for this bit on the output waveform when MOEN=0 and CcxEN changes from 0 to 1. 0: OCx/OCxN output is disabled 1: If CCxEN=1, the invalid level is output during the dead time (the specific level value is affected by the polarity configuration), and the idle level is output after the dead time
11	RMOS	R/W	Run Mode Off-state Configure Run mode means MOEN=1; disable means CcxEN=0; this bit describes the impact of different values for this bit on the output waveform when MOEN=1 and CcxEN changes from 0 to 1.

Field	Name	R/W	Description
			0: OCx/OCxN output is disabled 1: OCx/OCxN first outptus invalid level (the specific level value is affected by the polarity configuration)
12	BRKEN	R/W	Brake Function Enable 0: Disable 1: Enable Note: When the protection level is 1, this bit cannot be modified.
13	BRKPOL	R/W	Brake Polarity Configure 0: The brake input BRK is valid at low level 1: The brake input BRK is valid at high level Note: When the protection level is 1, this bit cannot be modified. Writing to this bit requires an APB clock delay before it can be used.
14	AOEN	R/W	Automatic Output Enable 0: MOEN can only be set to 1 by software 1: MOEN can be set to 1 by software or be automatically set to 1 in next update event (braking input is ineffective) Note: When the protection level is 1, this bit cannot be modified.
15	MOEN	R/W	PWM Main Output Enable 0: Disable the output of OCx and OCxN or force the output of idle state 1: When CCxEN and CCxNEN bits of the TMRx_CCEN register are set, turn on OCx and OCxN output When the brake input is valid, it is cleared by hardware asynchronously. Note: Setting to 1 by software or setting to 1 automatically depends on AOEN bit of the TMRx_BDT register.

15.6.19 DMA control register (TMRx_DCTRL)

Offset address: 0x48

Reset value: 0x0000

Field	Name	R/W	Description
4:0	DBADDR	R/W	DMA Base Address Setup These bits define the base address of DMA in continuous mode (when reading or writing TMRx_DMADDR register), and DBADDR is defined as the offset from the address of TMRx_CTRL1 register: 00000：TMRx_CTRL1 00001：TMRx_CTRL2 00010：TMRx_SMCTRL …….
7:5	Reserved		
12:8	DBLEN	R/W	DMA Burst Transfer Length Setup These bits define the transfer length and transfer times of DMA in continuous mode. The data transferred can be 16 bits and 8 bits. When reading/writing TMRx_DMADDR register, the timer will conduct a continuous transmission; 00000: Transmission once 00001: Transmission twice 00010: Transmission for three times ……

Field	Name	R/W	Description
			10001: Transmission for 18 times The transmission address formula is as follows: Transmission address=TMRx_CTRL1 address (slave address) +DBADDR+DMA index; DMA index=DBLEN For example: DBLEN=7, DBADDR=TMR1_CTRL1 (slave address) means the address of the data to be transmitted, while the address +DBADDR+7 of TMRx_CTRL1 means the address of the data to be written/read, Data transmission will occur to: TMRx_CTRL1 address + 7 registers starting from DBADDR. The data transmission will change according to different DMA data length: (1) When the transmission data is set to 16 bits, the data will be transmitted to seven registers (2) When the transmission data is set to 8 bits, the data of the first register is the MSB bit of the first data, the data of the second register is the LSB bit of the first data, and the data will still be transmitted to seven registers.
15:13	Reserved		

15.6.20 DMA address register of continuous mode (TMRx_DMADDR)

Offset address: 0x4C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	DMADDR	R/W	DMA Register for Burst Transfer Read or write operation access of TMRx_DMADDR register may lead to access operation of the register in the following address: TMRx_CTRL1 address + (DBADDR+DMA index) ×4 Wherein: "TMRx_CTRL1 address" is the address of control register 1 (TMRx_CTRL1); "DBADDR" is the base address defined in TMRx_DCTRL register; "DMA index" is the offset automatically controlled by DMA, and it depends on DBLEN defined in TMRx_DCTRL register.

16 General-purpose Timer (TMR2/3/4/5)

16.1 Introduction

The general-purpose timer takes the time base unit as the core, and has the functions of input capture and output compare, and can be used to measure the pulse width, frequency and duty cycle, and generate the output waveform. It includes a 16-bit auto reload counter (realize count-up, count-down and center-aligned count).

The timer and timer are independent of each other, and they can achieve synchronization and cascading.

16.2 Main Characteristics

（1） Timebase unit

- Counter: 16-bit counter, count-up, count-down and center-aligned count.
- Prescaler: 16-bit programmable prescaler
- Auto reloading function

（2） Clock source selection

- Internal clock
- External input
- External trigger
- Internal trigger

（3） Input capture function

- Counting function
- PWM input
- Encoder interface mode

（4） Output compare function

- PWM output mode
- Forced output mode
- Single-pulse mode

（5） Master/Slave mode controller of timer

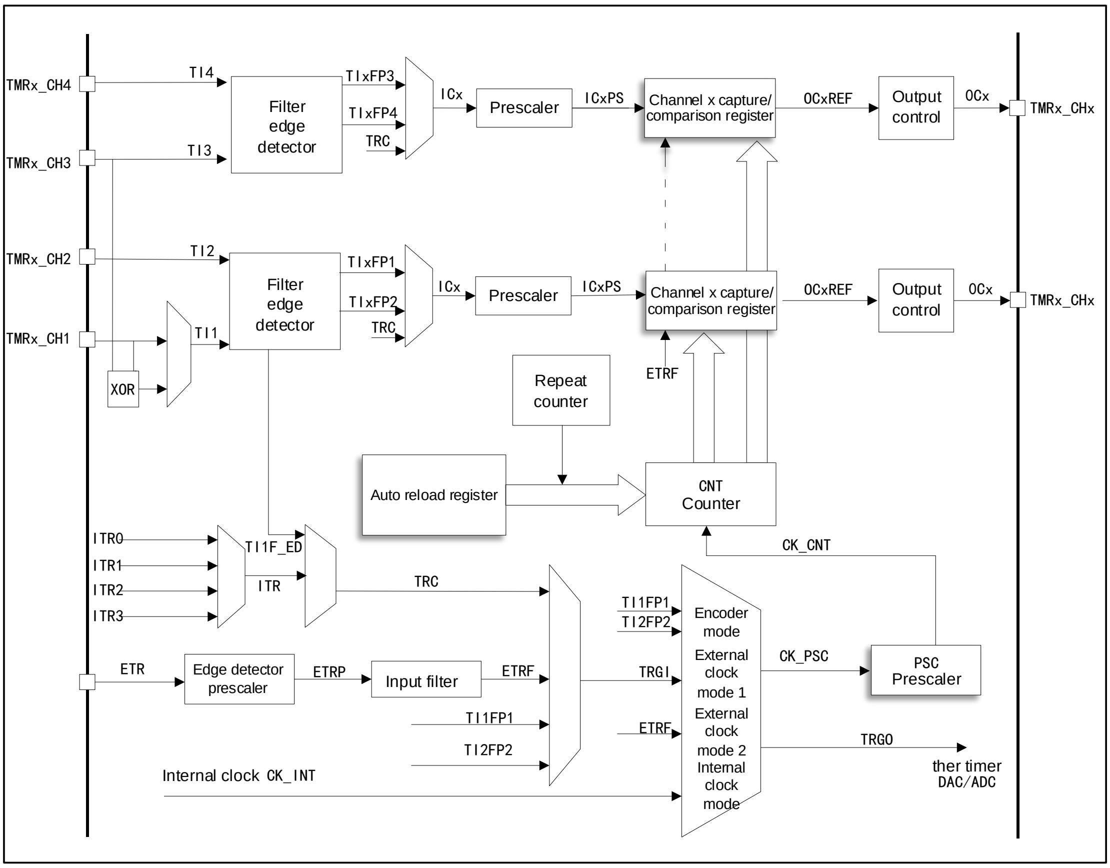
- Timers can be synchronized and cascaded
- Support multiple slave modes and synchronization signals

（6） Interrupt and DMA request event

- Update event (counter overrun/underrun, counter initialization)
- Trigger event (counter start, stop, internal/external trigger)
- Input capture
- Output compare

16.3 Structure Block Diagram

Figure 45 General-purpose Timer Structure Block Diagram



16.4 Functional Description

16.4.1 Clock Source Selection

The general-purpose timer has four clock sources.

Internal clock

It is TMRx_CLK from RCM, namely the driving clock of the timer; when the slave mode controller is disabled, the clock source CK_PSC of the prescaler is driven by the internal clock CK_INT.

External clock mode 1

The trigger signal generated from the input channel TI1/2/3/4 of the timer after polarity selection and filtering is connected to the slave mode controller to control the work of the counter. Besides, the pulse signal generated by the input of Channel 1 after double-edge detection of the rising edge and the falling edge is logically equal or the future signal is TI1F_ED signal, namely double-edge signal of TIF_ED. Specially the PWM input can only be input by TI1/2.

External clock mode 2

After polarity selection, frequency division and filtering, the signal from external trigger interface (ETR) is connected to slave mode controller through trigger input selector to control the work of counter.

Internal trigger input

The timer is set to work in slave mode, and the clock source is the output signal of other timers. At this time, the clock source has no filtering, and the synchronization or cascading between timers can be realized. The master mode timer can reset, start, stop or provide clock for the slave mode timer.

16.4.2 Timebase Unit

The time base unit in the general-purpose timer contains three registers

- Counter register (CNT) 16 bits
- Auto reload register (AUTORLD) 16 bits
- Prescaler (PSC) 16 bits

Counter CNT

There are three counting modes for the counter in the general-purpose timer

- Count-up mode
- Count-down mode
- Center-aligned mode

Count-up mode

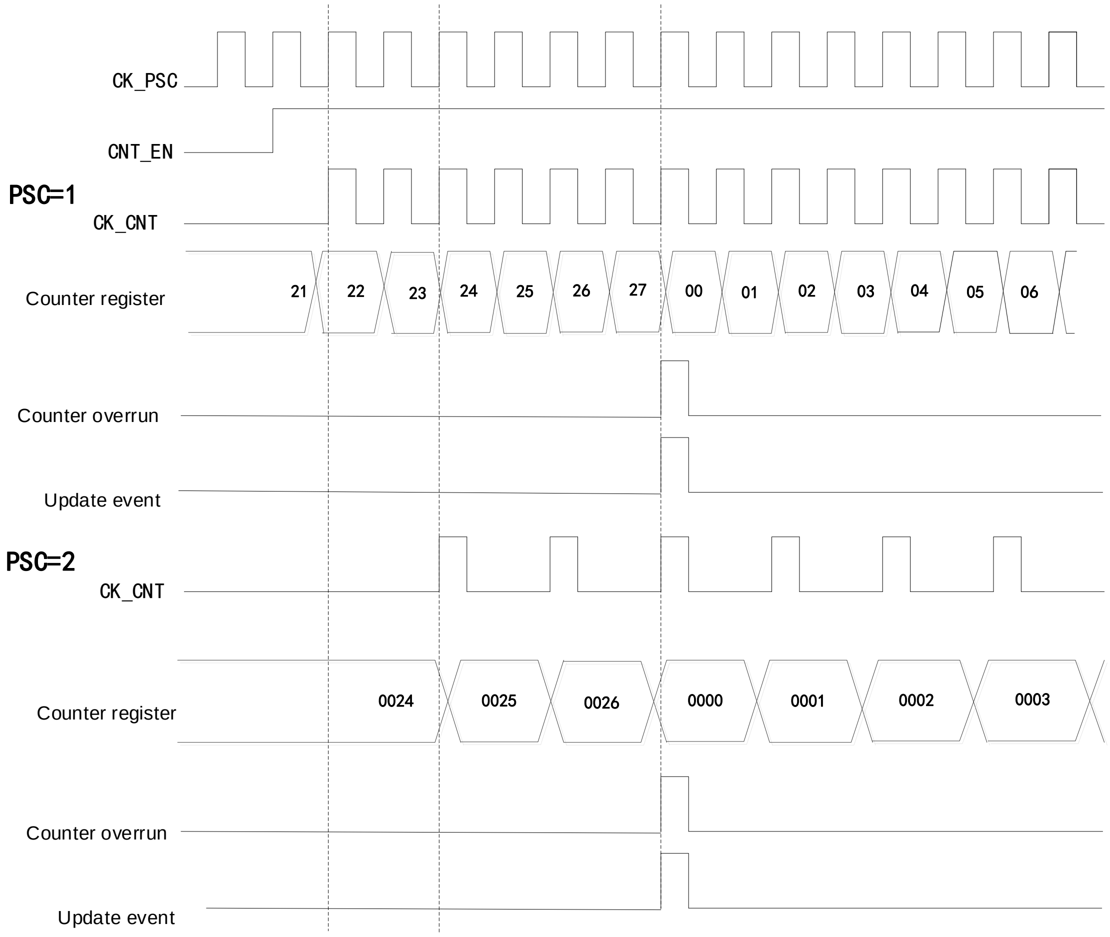
Set to the count-up mode by CNTDIR bit of configuration control register (TMRx_CTRL1).

When the counter is in count-up mode, the counter will count up from 0; every time a pulse is generated, the counter will increase by 1 and when the value of the counter (TMRx_CNT) is equal to the value of the auto reload (TMRx_AUTORLD), the counter will start to count again from 0, a count-up overrun event will be generated, and the value of the auto reload (TMRx_AUTORLD) is written in advance.

When the counter overruns, an update event will be generated. At this time, the auto reload shadow register and the prescaler buffer will be updated. The update event can be disabled by UD bit of configuration control register TMRx_CTRL1.

The figure below is Timing Diagram when Division Factor is 1 or 2 in Count-up Mode

Figure 46 Timing Diagram when Division Factor is 1 or 2 in Count-up Mode



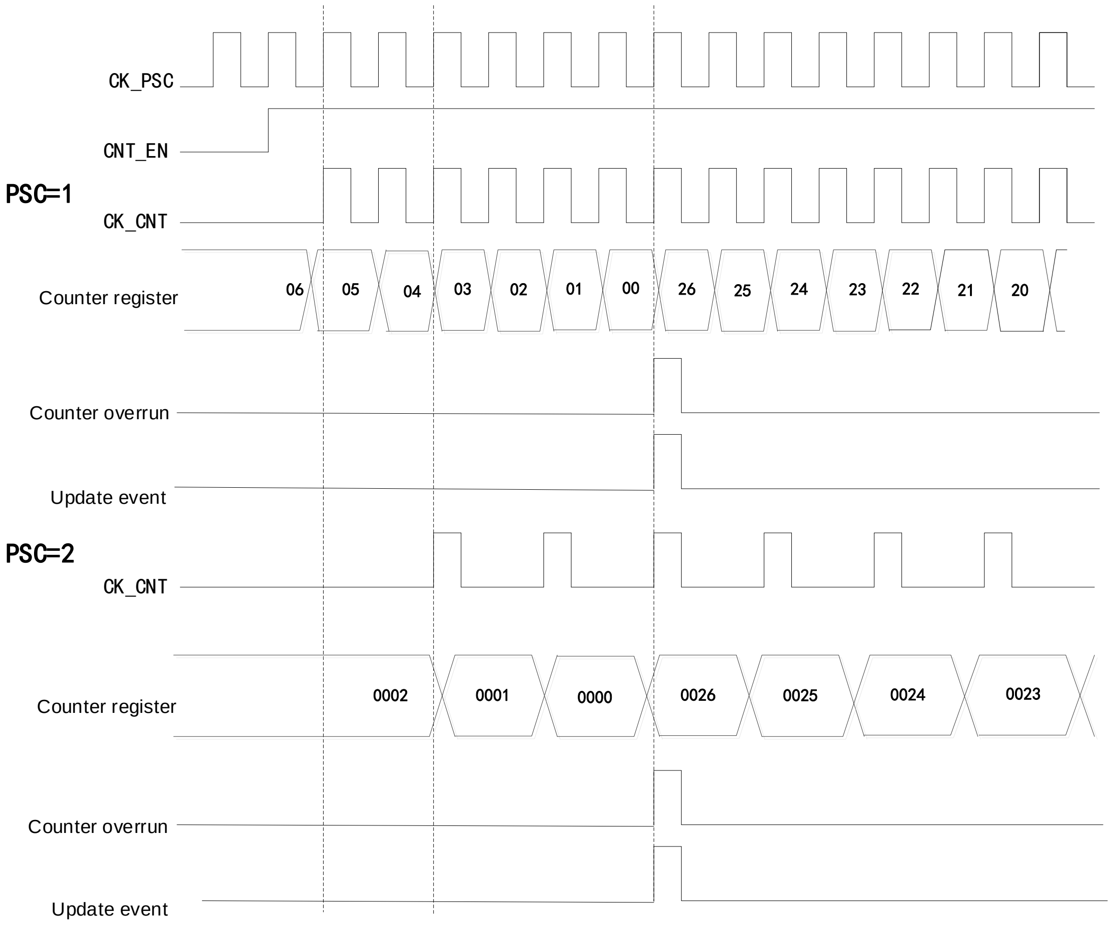
Count-down mode

Set to the count-down mode by CNTDIR bit of configuration control register (TMRx_CTRL1).

When the counter is in count-down mode, the counter will start to count down from the value of the auto reload (TMRx_AUTORLD); every time a pulse is generated, the counter will decrease by 1 and when it becomes 0, the counter will start to count again from (TMRx_AUTORLD), meanwhile, a count-down overrun event will be generated, and the value of the auto reload (TMRx_AUTORLD) is written in advance.

When the counter overruns, an update event will be generated. At this time, the auto reload shadow register and the prescaler buffer will be updated. The update event can be disabled by configuring the UD bit of the TMRx_CTRL1 register.

Figure 47 Timing Diagram when Division Factor is 1 or 2 in Count-down Mode

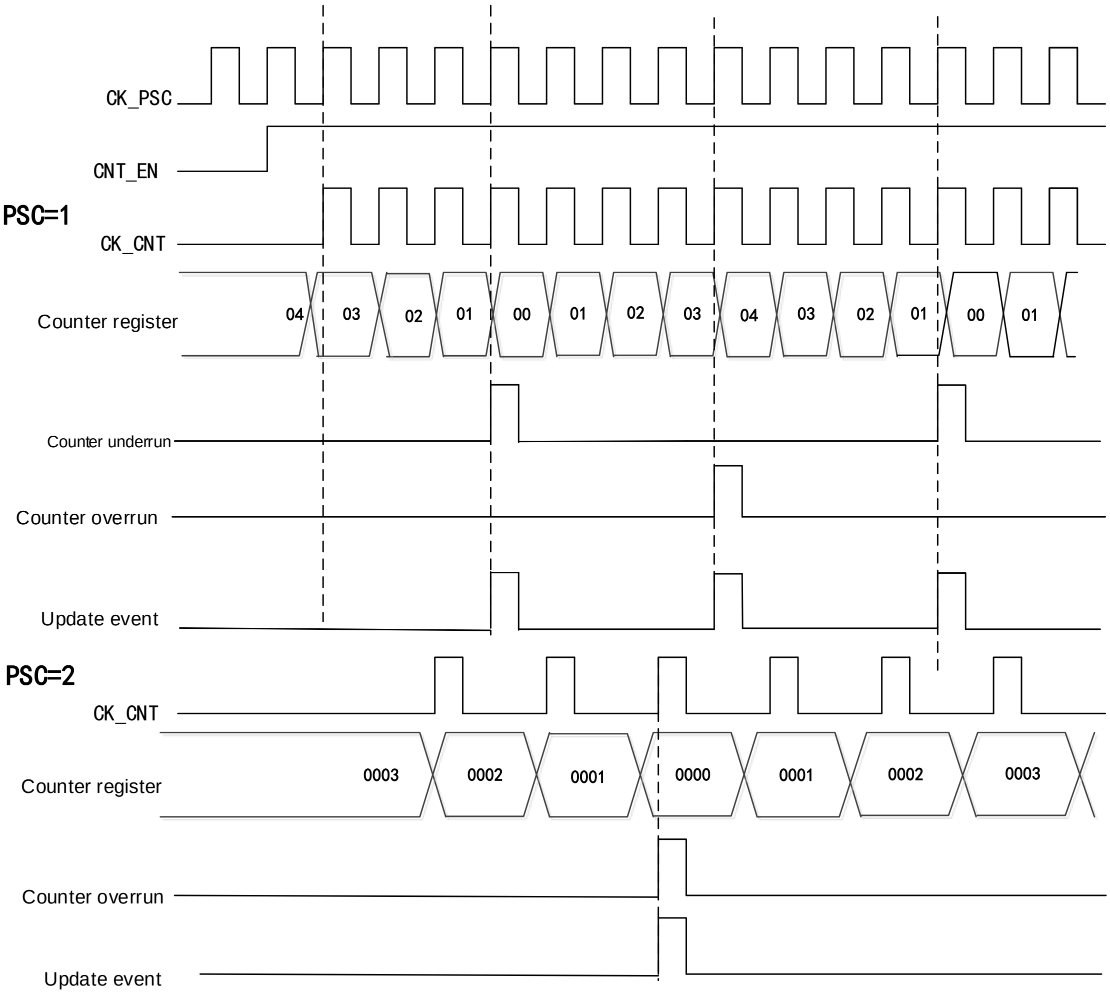


Center-aligned mode

Set to the center-aligned mode by CNTDIR bit of configuration control register (TMRx_CTRL1).

When the counter is in center-aligned mode, the counter counts up from 0 to the value of auto reload (TMRx_AUTORLD), then counts down to 0 from the value of the auto reload (TMRx_AUTORLD), which will repeat; in counting up, when the counter value is (AUTORLD-1), a counter overrun event will be generated; in counting down, when the counter value is 1, a counter underrun event will be generated.

Figure 48 Timing Diagram when Division Factor is 1 or 2 in Center-aligned Mode



Prescaler PSC

The prescaler is 16 bits and programmable, and it can divide the clock frequency of the counter to any value between 1 and 65536 (controlled by TMRx_PSC register), and after frequency division, the clock will drive the counter CNT to count. The prescaler has a buffer, which can be changed during running.

16.4.3 Input Capture

Input capture channel

The general-purpose timer has four independent capture/comparison channels, each of which is surrounded by a capture/comparison register.

In the input capture, the measured signal will enter from the external pin T1/2/3/4 of the timer, first pass through the edge detector and input filter, and then into the capture channel. Each capture channel has a corresponding capture register. When the capture occurs, the value of the counter CNT will be latched in the capture register CCx. Before entering the capture register, the signal will pass through the prescaler, which is used to set how many events to capture at a time.

Input capture application

Input capture is used to capture external events, and can give the time flag to indicate the occurrence time of the event and measure the pulse jump edge events (measure the frequency or pulse width), for example, if the selected edge appears on the input pin, the TMRx_CCx register will capture the current value of the counter and the CCxIFLG bit of the state register TMRx_STS will be set to 1; if CCxIEN=1, an interrupt will be generated.

In capture mode, the timing, frequency, period and duty cycle of a waveform can be measured. In the input capture mode, the edge selection is set to rising edge detection. When the rising edge appears on the capture channel, the first capture occurs, at this time, the value of the counter CNT will be latched in the capture register CCx; at the same time, it will enter the capture interrupt, a capture will be recorded in the interrupt service program and the value will be recorded. When the next rising edge is detected, the second capture occurs, the value of counter CNT will be latched in capture register CCx again, at this time, it will enter the capture interrupt again, the value of capture register will be read, and the cycle of this pulse signal will be obtained through capture.

16.4.4 Output Compare

There are eight modes of output compare: freeze, channel x is valid level when matching, channel x is invalid level when matching, flip, force is invalid, force is valid, PWM1 and PWM2 modes, which are configured by OCxMOD bit in TMRx_CCMx register and can control the waveform of output signal in output compare mode.

Output compare application

In the output compare mode, the position, polarity, frequency and time of the pulse generated by the timer can be controlled.

When the value of the counter is equal to that of the capture/comparison register, the channel output can be set as high level, low level or flip by configuring the OCxMOD bit in TMRx_CCMx register and the CCxPOL bit in the output polarity TMRx_CCEN register.

When CCxIFLG=1 in TMRx_STS register, if CCxIEN=1 in TMRx_DIEN register, an interrupt will be generated; if CCDSEL=1 in TMRx_CTRL2 register, DMA request will be generated.

16.4.5 PWM Output Mode

PWM mode is an adjustable pulse signal output by the timer. The pulse width of the signal is determined by the value of the comparison register CCx, and the cycle is determined by the value of the auto reload AUTORLD.

PWM output mode contains PWM mode 1 and PWM mode 2; PWM mode 1 and PWM mode 2 are divided into count-up, count-down and edge alignment counting; in PWM mode 1, if the value of the counter CNT is less than the value of the comparison register CCx, the output level will be valid; otherwise, it will be invalid.

Set the timing diagram in PWM1 mode when CCx=5, AUTORLD=7

Figure 49 PWM1 Count-up Mode Timing Diagram

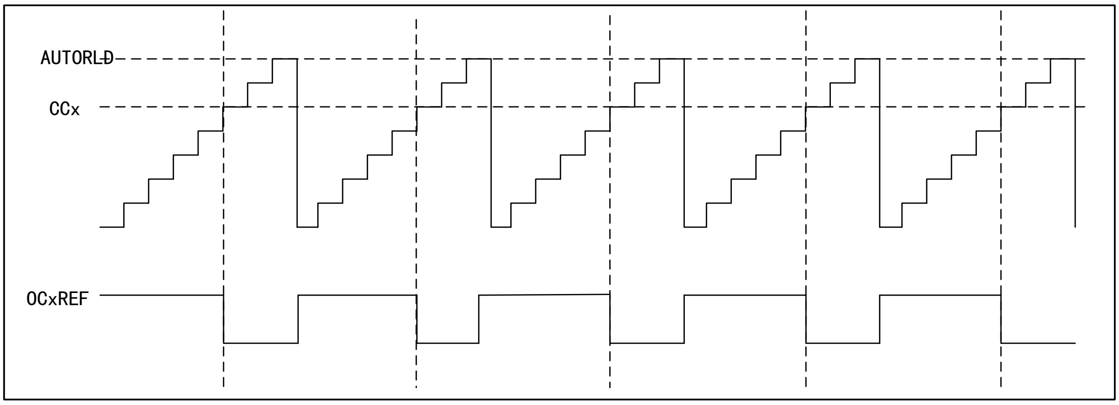


Figure 50 PWM1 Count-down Mode Timing Diagram

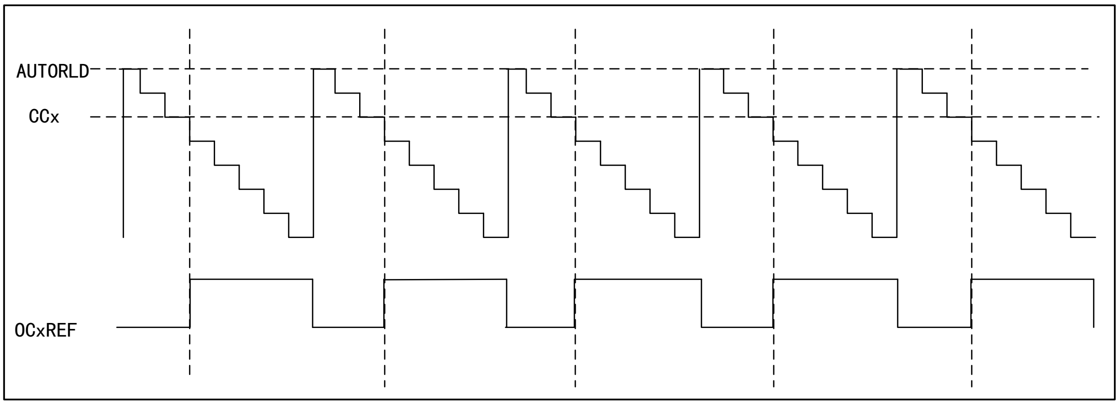
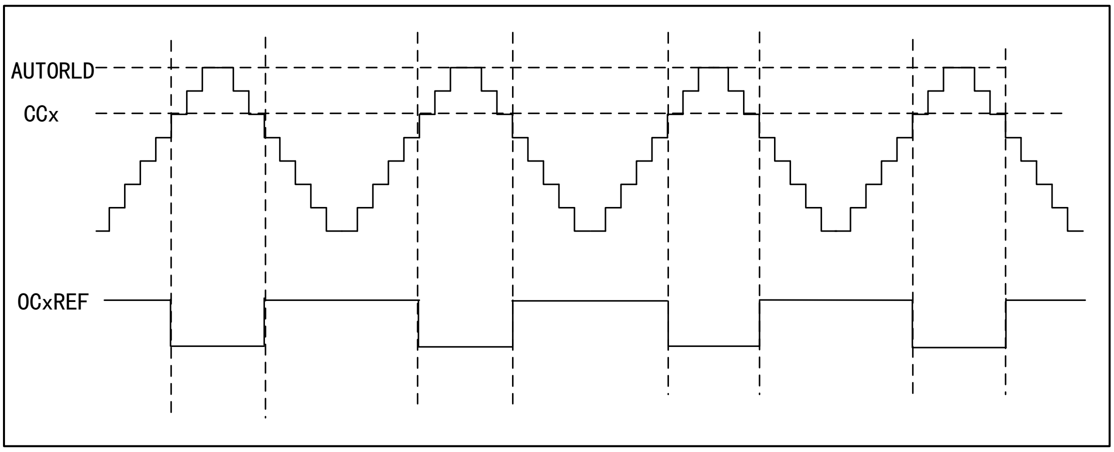


Figure 51 PWM1 Center-aligned Mode Timing Diagram



In PWM mode 2, if the value of the counter CNT is less than that of the comparison register CCx, the output level will be invalid; otherwise, it will be valid.

Set the timing diagram in PWM2 mode when CCx=5, AUTORLD=7

Figure 52 PWM2 Count-up Mode Timing Diagram

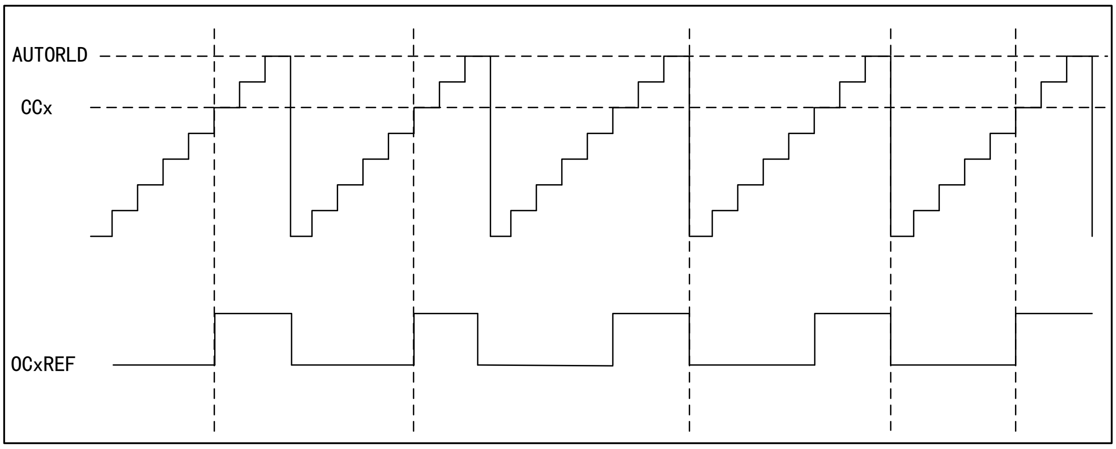


Figure 53 PWM2 Count-down Mode Timing Diagram

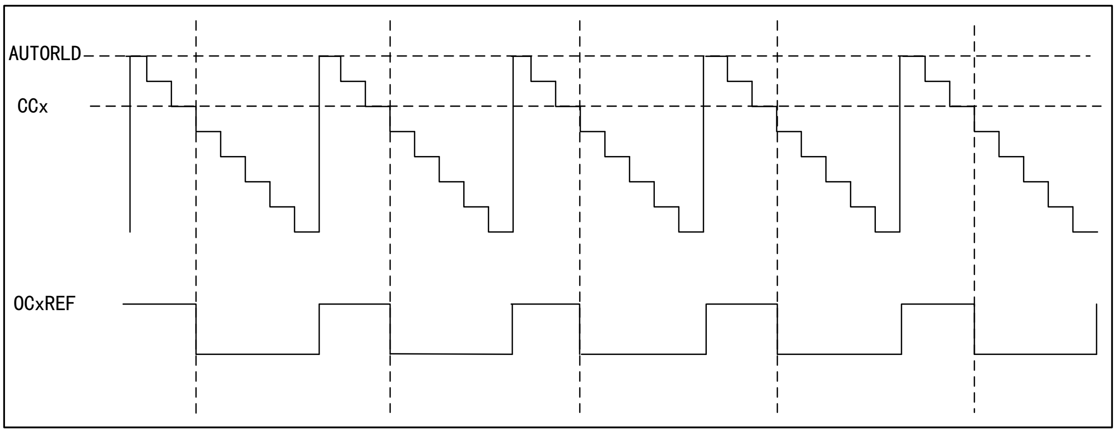
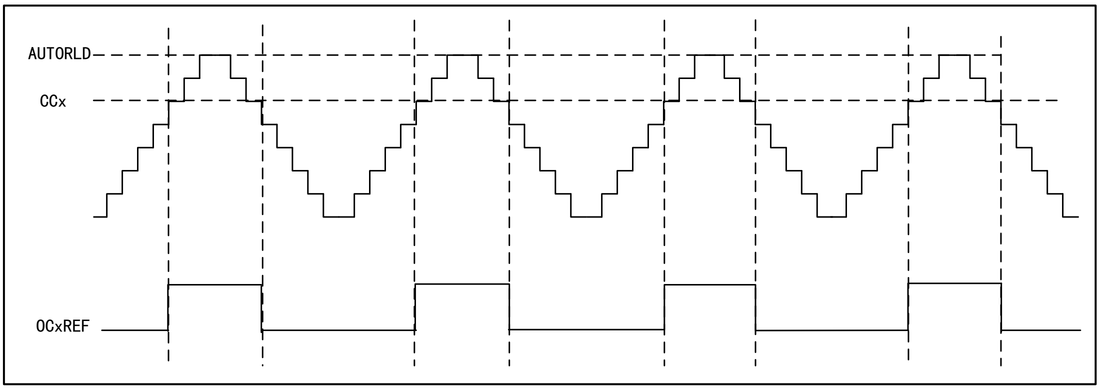


Figure 54 PWM2 Center-aligned Mode Timing Diagram



16.4.6 PWM Input Mode

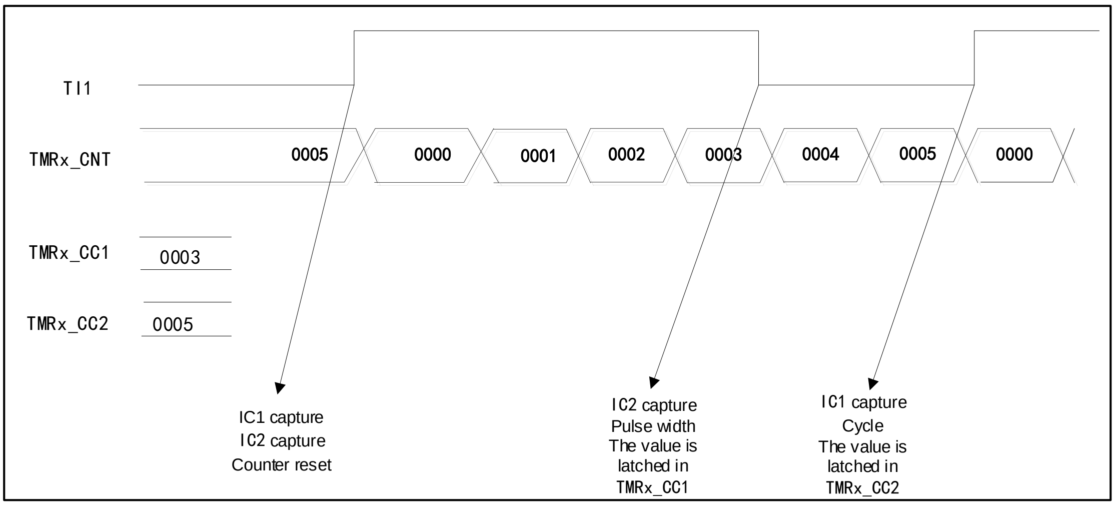
PWM input mode is a particular case of input capture.

In PWM input mode, as only TI1FP1 and TI1FP2 are connected to the slave mode controller, input can be performed only through the channels TMRx_CH1 and TMRx_CH2, which need to occupy the capure registers of CH1 and CH2.

In the PWM input mode, the PWM signal enters from TMRx_CH1, and the signal will be divided into two channels, one can measure the cycle and the other can measure the duty cycle. In the configuration, it is only required to set the polarity of one channel, and the other will be automatically configured with the opposite polarity.

In this mode, the slave mode controller should be configured as the reset mode (SMFSEL bit of TMRx_SMCTRL register)

Figure55 PWM Input Mode Timing Diagram



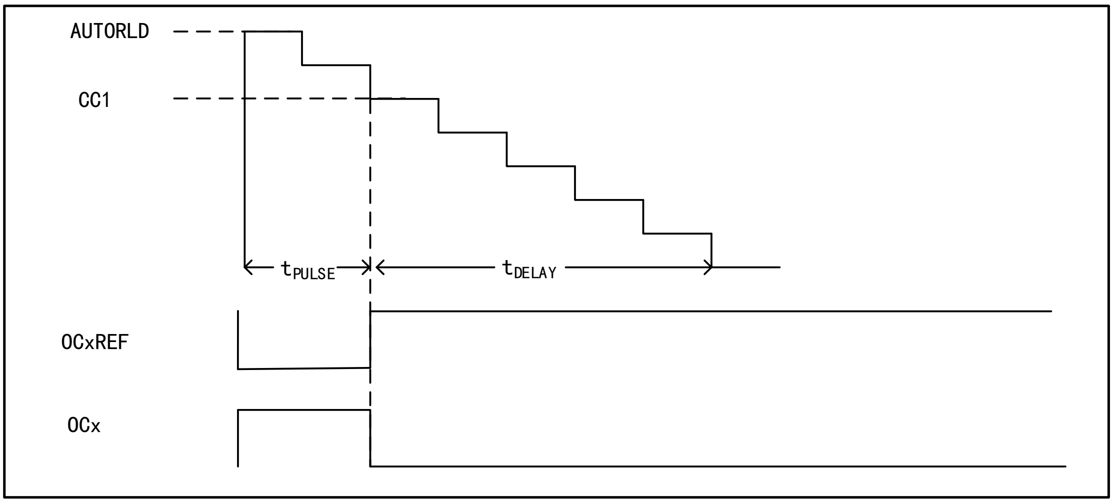
16.4.7 Single-pulse Mode

The single-pulse mode is a special case of timer comparison output, and is also a special case of PWM output mode.

Set SPMEN bit of TMRx_CTRL1 register, and select the single-pulse mode. After the counter is started, a certain number of pulses will be output before the update event occurs. When an update event occurs, the counter will stop counting, and the subsequent PWM waveform output will no longer be changed.

After a certain controllable delay, a pulse with controllable pulse width is generated in single-pulse mode through the program. The delay time is defined by the value of TMRx_CCx register; in the count-up mode, the delay time is CCx and the pulse width is AUTORLD-CCx; in the count-down mode, the delay time is AUTORLD-CCx and the pulse width is CCx.

Figure 56 Timing Diagram in Single-pulse Mode



16.4.8 Forced Output Mode

In the forced output mode, the comparison result is ignored, and the corresponding level is directly output according to the configuration instruction.

- CCxSEL=00 for TMRx_CCMx register, set CCx channel as output
- OCxMOD=100/101 for TMRx_CCMx register, set to force OCxREF signal to invalid/valid state

In this mode, the corresponding interrupt and DMA request will still be generated.

16.4.9 Encoder Interface Mode

The encoder interface mode is equivalent to an external clock with direction selection. In the encoder interface mode, the content of the timer can always indicate the position of the encoder.

The selection methods of encoder interface is as follows:

- By setting SMFSEL bit of TMRx_SMCTRL register, set the counter to count on the edge of TI1 channel /TI2 channel, or count on the edge of TI1 and TI2 at the same time.
- Select the polarity of TI1 and TI2 by setting the CC1POL and CC2POL bits of TMRx_CCEN register.
- Select to filter or not by setting the IC1F and IC2F bits of TMRx_CCM1 register.

The two input TI1 and TI2 can be used as the interface of incremental encoder. The counter is driven by the effective jump of the signals TI1FP1 and TI2FP2 after filtering and edge selection in TI1 and TI2.

The count pulse and direction signal are generated according to the input signals of TI1 and TI2

- The counter will count up/down according to the jumping sequence of the input signal

- Set CNTDIR of control register TMRx_CTRL1 to be read-only (CNTDIR will be re-calculated due to jumping of any input end)

The change mechanism of counter count direction is shown in the figure below

Table 72 Relationship between Count Direction and Encoder

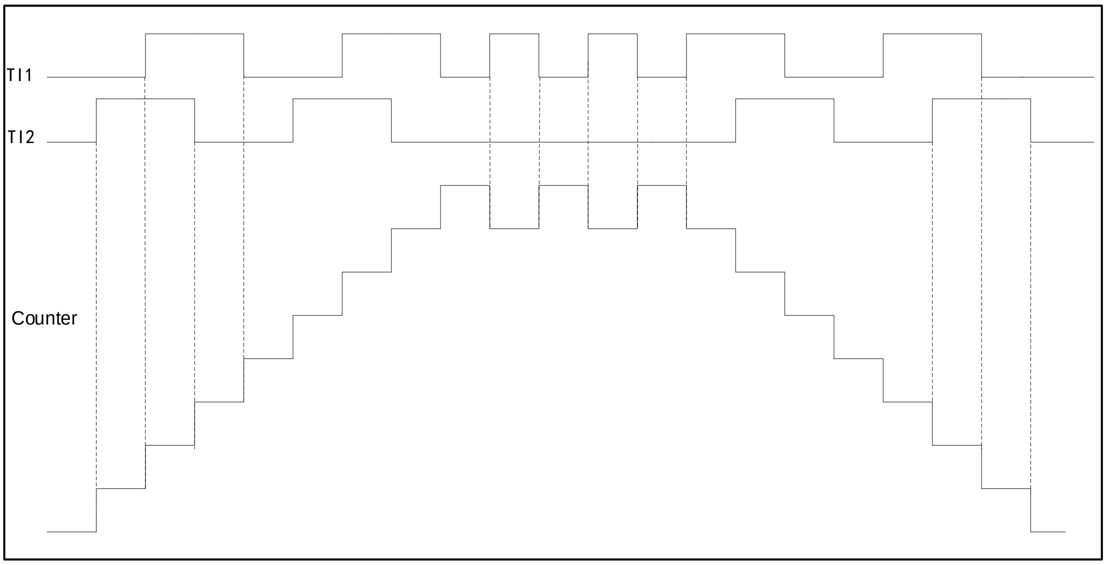
Effective edge		**Count only in TI1**		**Count only in TI2**		**Count in both TI1 and TI2**	
Level of relative signal		High	Low	High	Low	High	Low
TI1FP1	Rising edge	—		Count down	Count up	Count down	Count up
	Falling edge			Count up	Count down	Count up	Count down
TI2FP2	Rising edge	Count up	Count down	—		Count up	Count down
	Falling edge	Count down	Count up			Count down	Count up

The external incremental encoder can be directly connected with MCU, not needing external interface logic, so the comparator is used to convert the differential output of the encoder to digital signal to increase the immunity from noise interference.

Among the following examples,

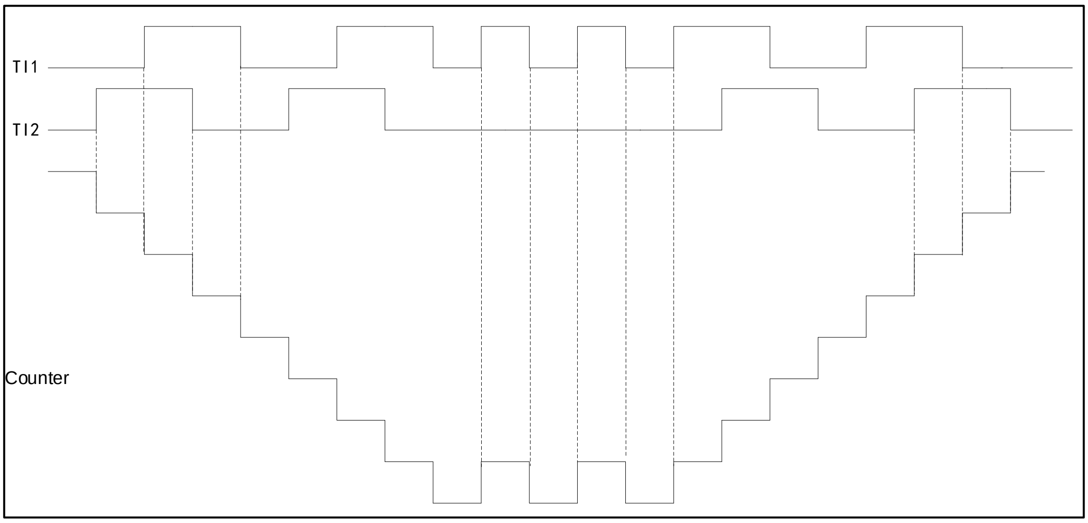
- IC1FP1 is mapped to TI1
- IC2FP2 is mapped to TI2
- Neither IC1FP1 nor IC2FP2 is reverse phase
- The input signal is valid at the rising edge and falling edge
- Enable the counter

Figure 57 Counter Operation Example in Encoder Mode



For example, when TI1 is at low level, and TI2 is in rising edge state, the counter will count up.

Figure 58 Example of Encoder Interface Mode of IC1FP1 Reversed Phase



For example, when TI1 is at low level, and the rising edge of TI2 jumps, the counter will count down.

16.4.10 Slave Mode

TMRx timer can synchronize external trigger

- Reset mode
- Gated mode
- Trigger mode

SMFSEL bit in TMRx_SMCTRL register can be set to select the mode

SMFSEL=100 set the reset mode, SMFSEL=101 set the gated mode, SMFSEL=110 set the trigger mode.

In the reset mode, when a trigger input event occurs, the counter and prescaler will be initialized, and the rising edge of the selected trigger input (TRGI) will reinitialize the counter and generate a signal to update the register.

In the gated mode, the enable of the counter depends on the high level of the selected input. When the trigger input is high, the clock of the counter will be started. Once the trigger input becomes low, the counter will stop (but not be reset). The start and stop of the counter are controlled.

In the trigger mode, the enable of the counter depends on the event on the selected input, the counter is started (but is not reset) at the rising edge of the trigger input, and only the start of the counter is controlled.

16.4.11 Timer Interconnection

See timer interconnection for details

16.4.12 Interrupt and DMA Request

The timer can generate an interrupt when an event occurs during operation

- Update event (counter overrun/underrun, counter initialization)
- Trigger event (counter start, stop, internal/external trigger)
- Capture/Comparison event

Some internal interrupt events can generate DMA requests, and special interfaces can enable or disable DMA requests.

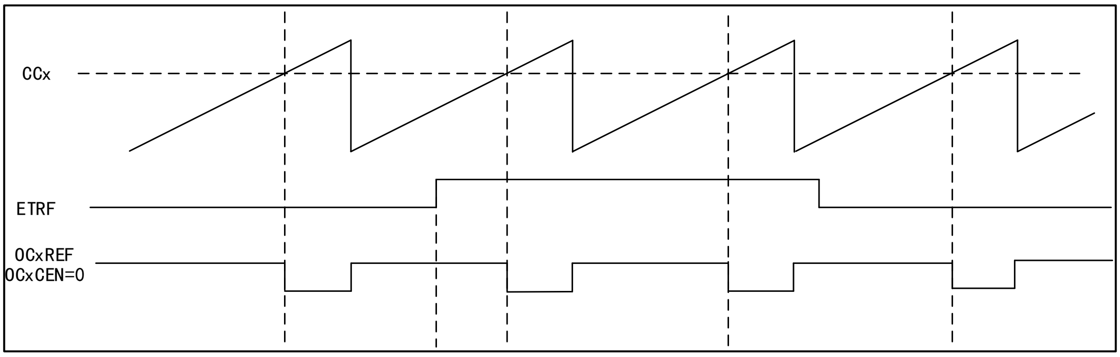
16.4.13 Clear OCxREF Signal when External Events Occur

This function is used for output compare and PWM mode.

In one channel, the high level of ETRF input port will reduce the signal of OCxREF to low level, and the OCxCEN bit in capture/comparison register TMRx_CCMx is set to 1, and OCxREF signal will remain low until the next update event.

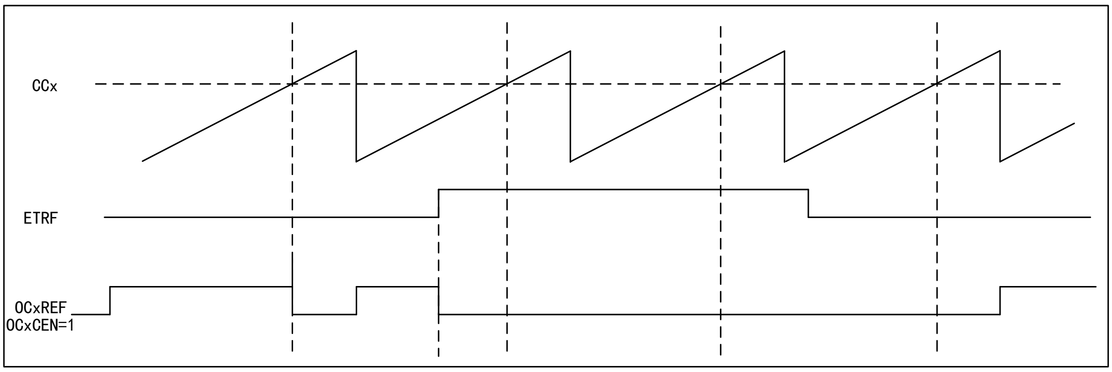
Set TMRx to PWM mode, close the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=0, and the output OCxREF signal is shown in the figure below.

Figure 59 OCxREF Timing Diagram



Set TMRx to PWM mode, close the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=1, and the output OCxREF signal is shown in the figure below.

Figure 60 OCxREF Timing Diagram



16.5 Register Address Mapping

In the following table, all registers of the general-purpose timer are mapped to a 16-bit addressable (address) space.

Table 73 General-purpose Timer Register Address Mapping

Register name	Description	Offset address
TMRx_CTRL1	Control register 1	0x00
TMRx_CTRL2	Control register 2	0x04
TMRx_SMCTRL	Slave mode control register	0x08
TMRx_DIEN	DMA/Interrupt enable register	0x0C
TMRx_STS	State register	0x10
TMRx_CEG	Control event generation register	0x14
TMRx_CCM1	Capture/Comparison mode register 1	0x18
TMRx_CCM2	Capture/Comparison mode register 2	0x1C
TMRx_CCEN	Capture/Comparison enable register	0x20
TMRx_CNT	Counter register	0x24
TMRx_PSC	Prescaler register	0x28
TMRx_AUTORLD	Auto reload register	0x2C
TMRx_CC1	Channel 1 capture/comparison register	0x34
TMRx_CC2	Channel 2 capture/comparison register	0x38
TMRx_CC3	Channel 3 capure/comparison register	0x3C
TMRx_CC4	Channel 4 capture/comparison register	0x40
TMRx_DCTRL	DMA control register	0x48
TMRx_DMADDR	DMA address register of continuous mode	0x4C

16.6 Register Functional Description

16.6.1 Control register 1 (TMRx_CTRL1)

Offset address: 0x00

Reset value: 0x0000

Field	Name	R/W	Description
0	CNTEN	R/W	Counter Enable 0: Disable 1: Enable When the timer is configured as external clock, gated mode and encoder mode, it is required to write 1 to the bit by software to start regular work;

Field	Name	R/W	Description
			when it is configured as the trigger mode, it can be written to 1 by hardware.
1	UD	R/W	Update Disable Update event can cause AUTORLD, PSC and CCx to generate the value of update setting. 0: Update event is allowed (UEV) An update event can occur in any of the following situations: The counter overruns/underruns; Set UEG bit; Update generated by slave mode controller. 1: Update event is disabled
2	URSSEL	R/W	Update Request Source Select If interrupt or DMA is enabled, the update event can generate update interrupt or DMA request. Different update request sources can be selected through this bit. 0: The counter overruns or underruns Set UEG bit Update generated by slave mode controller 1: The counter overruns or underruns
3	SPMEN	R/W	Single Pulse Mode Enable When an update event is generated, the output level of the channel can be changed; in this mode, the CNTEN bit will be cleared, the counter will be stopped, and the output level of the channel will not be changed. 0: Disable 1: Enable
4	CNTDIR	R/W	Counter Direction When the counter is configured in central alignment mode or encoder mode, the bit is read-only. 0: Count up 1: Count down
6:5	CAMSEL	R/W	Center Aligned Mode Select In the center-aligned mode, the counter counts up and down alternately; otherwise, it will only count up or down. Different center-aligned modes affect the timing of setting the output compare interrupt flag bit of the output channel to 1; when the counter is disabled (CNTEN=0), select the center-aligned mode. 00: Edge alignment mode 01: Center-aligned mode 1 (the output compare interrupt flag bit of output channel is set to 1 when counting down) 10: Center-aligned mode 2 (the output compare interrupt flag bit of output channel is set to 1 when counting up) 11: Center-aligned mode 3 (the output compare interrupt flag bit of output channel is set to 1 when counting up/down)
7	ARPEN	R/W	Auto-reload Preload Enable When the buffer is disabled, the program modification TMRx_AUTORLD will immediately modify the values loaded to the counter; when the buffer is enabled, the program modification TMRx_AUTORLD will modify the values loaded to the counter in the next update event. 0: Disable

Field	Name	R/W	Description
			1: Enable
9:8	CLKDIV	R/W	Clock Divide Factor For the configuration of dead time and digital filter, CK_INT provides the clock, and the dead time and the clock of the digital filter can be adjusted by setting this bit. 00: t_{DTS}=t_{CK_INT} 01: t_{DTS}=2×t_{CK_INT} 10: t_{DTS}=4×t_{CK_INT} 11: Reserved
15:10	Reserved		

16.6.2 Control register 2 (TMRx_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
2:0	Reserved		
3	CCDSEL	R/W	Capture/compare DMA Select 0: Send DMA request of CCx when CCx event occurs 1: Send DMA request of CCx when an update event occurs
6:4	MMSEL	R/W	Master Mode Signal Select The signals of timers working in master mode can be used for TRGO, which affects the work of timers in slave mode and cascaded with master timer, and specifically affects the configuration of timers in slave mode. 000: Reset; the reset signal of master mode timer is used for TRGO 001: Enable; the counter enable signal of master mode timer is used for TRGO 010: Update; the update event of master mode timer is used for TRGO 011: Comparison pulses; when the master mode timer captures/compares successfully (CCxIFLG=1), a pulse signal is output for TRGO 100: Comparison mode 1; OC1REF is used to trigger TRGO 101: Comparison mode 2; OC2REF is used to trigger TRGO 110: Comparison mode 3; OC3REF is used to trigger TRGO 111: Comparison mode 4; OC4REF is used to trigger TRGO
7	TI1SEL	R/W	Timer Input 1 Selection 0: TMRx_CH1 pin is connected to TI1 input 1: TMRx_CH1, TMRx_CH2 and TMRx_CH3 pins are connected to TI1 input after exclusive
15:8	Reserved		

16.6.3 Slave mode control register (TMRx_SMCTRL)

Offset address: 0x08

Reset value: 0x0000

Field	Name	R/W	Description
2:0	SMFSEL	R/W	Slave Mode Function Select 000: Disable the slave mode, the timer can be used as master mode timer to affect the work of slave mode timer; if CTRL1_CNTEN=1, the prescaler is directly driven by the internal clock. 001: Encoder mode 1; according to the level of TI1FP1, the counter counts at the edge of TI2FP2. 010: Encoder mode 2; according to the level of TI2FP2, the counter counts at the edge of TI1FP1. 011: Encoder mode 3; according to the input level of another signal, the counter counts at the edge of TI1FP1 and TI2FP2. 100: Reset mode; the slave mode timer resets the counter after receiving the rising edge signal of TRGI and generates the signal to update the register. 101: Gated mode; the slave mode timer starts the counter to work after receiving the TRGI high level signal; it stops the counter when receiving TRGI low level; when receiving TRGI high level signal again, the timer will continue to work; the counter is not reset during the whole period. 110: Trigger mode, the slave mode timer starts the counter to work after receiving the rising edge signal of TRGI. 111: External clock mode 1; select the rising edge signal of TRGI as the clock source to drive the counter to work.
3	Reserved		
6:4	TRGSEL	R/W	Trigger Input Signal Select In order to avoid false edge detection when changing the bit value, it must be changed when SMFSEL=0. 000: Internal trigger ITR0 001: Internal trigger ITR1 010: Internal trigger ITR2 011: Internal trigger ITR3 100: Channel 1 input edge detector TIF_ED 101: Channel 1 post-filtering timer input TI1FP1 110: Channel 2 post-filtering timer input TI2FP2 111: External trigger input (ETRF)
7	MSMEN	R/W	Master/slave Mode Enable 0: Invalid 1: Enable the master/slave mode
11:8	ETFCFG	R/W	External Trigger Filter Configure 0000: Filter disabled, sampling by f_{DTS} 0001: DIV=1, N=2 0010: DIV=1, N=4 0011: DIV=1, N=8 0100: DIV=2, N=6 0101: DIV=2, N=8 0110: DIV=4, N=6 0111: DIV=4, N=8 1000: DIV=8, N=6 1001: DIV=8, N=8

Field	Name	R/W	Description
			1010: DIV=16, N=5 1011: DIV=16, N=6 1100: DIV=16, N=8 1101: DIV=32, N=5 1110: DIV=32, N=6 1111: DIV=32, N=8 Sampling frequency=timer clock frequency/DIV; the filter length=N, and a jump is generated by every N events.
13:12	ETPCFG	R/W	External Trigger Prescaler Configure The ETR (external trigger input) signal becomes ETRP after frequency division. The signal frequency of ETRP is at most 1/4 of TMRxCLK frequency; when ETR frequency is too high, the ETRP frequency must be reduced through frequency division. 00: The prescaler is disabled; 01: ETR signal 2 divided frequency 10: ETR signal 4 divided frequency 11: ETR signal 8 divided frequency
14	ECEN	R/W	External Clock Mode2 Enable 0: Disable 1: Enable Setting ECEN bit has the same function as selecting external clock mode 1 to connect TRG1 to ETRF; slave mode (reset, gating, trigger) can be used at the same time with external clock mode 2, but TRGI cannot be connected to ETRF in such case; when external clock mode 1 and external clock mode 2 are enabled at the same time, the input of external clock is ETRF.
15	ETPOL	R/W	External Trigger Polarity Configure This bit decides whether the external trigger ETR is reversed. 0: The external trigger ETR is not reversed,and the high level or rising edge is valid 1: The external trigger ETR is reversed, and the low level or falling edge is valid

Table 74 TMRx Internal Trigger Connection

Slave timer	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TMR2	TMR1	TMR8	TMR3	TMR4
TMR3	TMR1	TMR2	TMR5	TMR4
TMR4	TMR1	TMR2	TMR3	TMR8
TMR5	TMR2	TMR3	TMR4	TMR8

16.6.4 DMA/Interrupt enable register (TMRx_DIEN)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
0	UIEN	R/W	Update interrupt Enable 0: Disable

Field	Name	R/W	Description
			1: Enable
1	CC1IEN	R/W	Capture/Compare Channel1 Interrupt Enable 0: Disable 1: Enable
2	CC2IEN	R/W	Capture/Compare Channel2 Interrupt Enable 0: Disable 1: Enable
3	CC3IEN	R/W	Capture/Compare Channel3 Interrupt Enable 0: Disable 1: Enable
4	CC4IEN	R/W	Capture/Compare Channel4 Interrupt Enable 0: Disable 1: Enable
5	Reserved		
6	TRGIEN	R/W	Trigger interrupt Enable 0: Disable 1: Enable
7	Reserved		
8	UDIEN	R/W	Update DMA Request Enable 0: Disable 1: Enable
9	CC1DEN	R/W	Capture/Compare Channel1 DMA Request Enable 0: Disable 1: Enable
10	CC2DEN	R/W	Capture/Compare Channel2 DMA Request Enable 0: Disable 1: Enable
11	CC3DEN	R/W	Capture/Compare Channel3 DMA Request Enable 0: Disable 1: Enable
12	CC4DEN	R/W	Capture/Compare Channel4 DMA Request Enable 0: Disable 1: Enable
13	Reserved		
14	TRGDEN	R/W	Trigger DMA Request Enable 0: Disable 1: Enable
15	Reserved		

16.6.5 State register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
0	UIFLG	RC_W0	Update Event Interrupt Generate Flag 0: Update event interrupt does not occur 1: Update event interrupt occurs When the counter value is reloaded or reinitialized, an update event will be generated. The bit is set to 1 by hardware and cleared by software; update events are generated in the following situations: (1) UD=0 on TMRx_CTRL1 register, and when the value of the repeat counter overruns/underruns, an update event will be generated; (2) URSEL=0 and UD=0 on TMRx_CTRL1 register, configure UG = 1 on TMRx_CEG register to generate update event, and the counter needs to be initialized by software; (3) URSEL=0 and UD=0 on TMRx_CTRL1 register, generate update event when the counter is initialized by trigger event.
1	CC1IFLG	RC_W0	Capture/Compare Channel1 Interrupt Flag **When the capture/comparison channel 1 is configured as output:** 0: No matching occurred 1: The value of TMRx_CNT matches the value of TMRx_CC1 **When the capture/comparison channel 1 is configured as input:** 0: Input capture did not occur 1: Input capture occurred When capture event occurs, the bit is set to 1 by hardware, and it can be cleared by software or cleared when reading TMRx_CC1 register.
2	CC2IFLG	RC_W0	Capture/Compare Channel2 new Interrupt Flag Refer to STS_CC1IFLG
3	CC3IFLG	RC_W0	Capture/Compare Channel3 Interrupt Flag Refer to STS_CC1IFLG
4	CC4IFLG	RC_W0	Captuer/Compare Channel4 Interrupt Flag Refer to STS_CC1IFLG
5	Reserved		
6	TRGIFLG	RC_W0	Trigger Event Interrupt Generate Flag 0: Trigger event interrupt did not occur 1: Trigger event interrupt occurred After Trigger event is generated, this bit is set to 1 by hardware and cleared by software.
8:7	Reserved		
9	CC1RCFLG	RC_W0	Capture/compare Channel1 Repetition Capture Flag 0: Repeat capture does not occur 1: Repeat capture occurs The value of the counter is captured to TMRx_CC1 register, and CC1IFLG=1; this bit is set to 1 by hardware and cleared by software only when the channel is configured as input capture.

Field	Name	R/W	Description
10	CC2RCFLG	RC_W0	Capture/compare Channel2 Repetition Capture Flag Refer to STS_CC1RCFLG
11	CC3RCFLG	RC_W0	Capture/compare Channel3 Repetition Capture Flag Refer to STS_CC1RCFLG
12	CC4RCFLG	RC_W0	Capture/compare Channel4 Repetition Capture Flag Refer to STS_CC1RCFLG
15:13	Reserved		

16.6.6 Control event generation register (TMRx_CEG)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
0	UEG	W	Update Event Generate 0: Invalid 1: Initialize the counter and generate the update event This bit is set to 1 by software, and cleared by hardware. Note: When an update event is generated, the counter of the prescaler will be cleared, but the prescaler factor remains unchanged. In the count-down mode, the counter reads the value of TMRx_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared.
1	CC1EG	W	Capture/Compare Channel1 Event Generation 0: Invalid 1: Capture/Comparison event is generated This bit is set to 1 by software and cleared automatically by hardware. If Channel 1 is in output mode When CC1IFLG=1, if CC1IEN and CC1DEN bits are set, the corresponding interrupt and DMA request will be generated. If Channel 1 is in input mode The value of the capture counter is stored in TMRx_CC1 register; configure CC1IFLG=1, and if CC1IEN and CC1DEN bits are also set, the corresponding interrupt and DMA request will be generated; at this time, if CC1IFLG=1, it is required to configure CC1RCFLG=1.
2	CC2EG	W	Capture/Compare Channel2 Event Generation Refer to CC1EG description
3	CC3EG	W	Capture/Compare Channel3 Event Generation Refer to CC1EG description
4	CC4EG	W	Capture/Compare Channel4 Event Generation Refer to CC1EG description
5	Reserved		
6	TEG	W	Trigger Event Generate 0: Invalid 1: Trigger event is generated This bit is set to 1 by software and cleared automatically by hardware.
15:7	Reserved		

16.6.7 Capature/Comparison mode register 1 (TMRx_CCM1)

Offset address: 0x18

Reset value: 0x0000

The timer can be configured as input (capture mode) or output (comparison mode) by CCxSEL bit. The functions of other bits of the register are different in input and output modes, and the functions of the same bit are different in output mode and input mode. The OCX in the register describes the function of the channel in the output mode, and the ICx in the register describes the function of the channel in the input mode.

Output compare mode:

Field	Name	R/W	Description
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Selection This bit defines the input/output direction and the selected input pin. 00: CC1 channel is output 01: CC1 channel is input, and IC1 is mapped on TI1 10: CC1 channel is input, and IC1 is mapped on TI2 11: CC1 channel is input, and IC1 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC1EN=0).
2	OC1FEN	R/W	Output Compare Channel1 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/comparison output to the trigger input event.
3	OC1PEN	R/W	Output Compare Channel1 Preload Enable 0: Preloading function is disabled; write the value of TMRx_CC1 register through the program and it will work immediately. 1: Preloading function is enabled; write the value of TMRx_CC1 register through the program and it will work after an update event is generated. Note: When the protection level is 3 and the channel is configured as output, this bit cannot be modified. When the preload register is uncertain, PWM mode can be used only in single pulse mode (SPMEN=1); otherwise, the following output compare result is uncertain.

Field	Name	R/W	Description
6:4	OC1MOD	R/W	Output Compare Channel1 Mode Configure 000: Freeze. The output compare has no effect on OC1REF 001: The output value is high when matching. When the value of counter CNT matches the value CCx of capture/compare register, OC1REF will be forced to be at high level 010: The output value is low when matching. When the value of the counter matches the value of the capture/compare register, OC1REF will be forced to be at low level 011: Output flaps when matching. When the value of the counter matches the value of the capture/compare register, flap the level of OC1REF 100: The output is forced to be ow Force OC1REF to be at low level 101: The output is forced to be high. Force OC1REF to be at high level 110: PWM mode 1 (set to high when the counter value<output compare value; otherwise, set to low) 111: PWM mode 2 (set to high when the counter value>output compare value; otherwise, set to low) Note: When the protection level is 3 and the channel is configured as output, this bit cannot be modified. In PWM modes 1 and 2, the OC1REF level changes when the comparison result changes or when the output compare mode changes from freeze mode to PWM mode.
7	OC1CEN	R/W	Output Compare Channel1 Clear Enable 0: OC1REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0
9:8	CC2SEL	R/W	Capture/Compare Channel2 Select This bit defines the input/output direction and the selected input pin. 00: CC2 channel is output 01: CC2 channel is input, and IC2 is mapped on TI2 10: CC2 channel is input, and IC2 is mapped on TI1 11: CC2 channel is input, and IC2 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC2EN=0).
10	OC2FEN	R/W	Output Compare Channel2 Preload Enable
11	OC2PEN	R/W	Output Compare Channel2 Buffer Enable
14:12	OC2MOD	R/W	Output Compare Channel1 Mode
15	OC2CEN	R/W	Output Compare Channel2 Clear Enable

Input capture mode:

Field	Name	R/W	Description
1:0	CC1SEL	R/W	Capture/Compare Channel 1 Select 00: CC1 channel is output 01: CC1 channel is input, and IC1 is mapped on TI1 10: CC1 channel is input, and IC1 is mapped on TI2 11: CC1 channel is input, and IC1 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC1EN=0).

Field	Name	R/W	Description
3:2	IC1PSC	R/W	Input Capture Channel 1 Perscaler Configure 00：PSC=1 01：PSC=2 10：PSC=4 11：PSC=8 PSC is prescaled factor, which triggers capture once every PSC events.
7:4	IC1F	R/W	Input Capture Channel 1 Filter Configuration 0000: Filter disabled, sampling by f_{DTS} 0001: DIV=1, N=2 0010: DIV=1, N=4 0011: DIV=1, N=8 0100: DIV=2, N=6 0101: DIV=2, N=8 0110: DIV=4, N=6 0111: DIV=4, N=8 1000: DIV=8, N=6 1001: DIV=8, N=8 1010: DIV=16, N=5 1011: DIV=16, N=6 1100: DIV=16, N=8 1101: DIV=32, N=5 1110: DIV=32, N=6 1111: DIV=32, N=8 Sampling frequency=timer clock frequency/DIV; the filter length=N, indicating that a jump is generated by every N events.
9:8	CC2SEL	R/W	Capture/Compare Channel 2 Select 00: CC2 channel is output 01: CC2 channel is input, and IC2 is mapped on TI1 10: CC2 channel is input, and IC2 is mapped on TI2 11: CC2 channel is input, and IC2 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC2EN=0).
11:10	IC2PSC	R/W	Input Capture Channel 2 Perscaler Configuration
15:12	IC2F	R/W	Input Capture Channel 2 Filter Configuration

16.6.8 Capture/Compare mode register 2 (TMRx_CCM2)

Offset address: 0x1C

Reset value: 0x0000

Refer to the description of the above CCM1 register.

Output compare mode:

Field	Name	R/W	Description
1:0	CC3SEL	R/W	Capture/Compare Channel 1 Selection This bit defines the input/output direction and the selected input pin. 00: CC3 channel is output

Field	Name	R/W	Description
			01: CC3 channel is input, and IC3 is mapped on TI3 10: CC3 channel is input, and IC3 is mapped on TI4 11: CC3 channel is input, and IC3 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC3EN=0).
2	OC3FEN	R/W	Output Compare Channel3 Fast Enable 0: Disable 1: Enable This bit is used to improve the response of the capture/compare output to the trigger input event.
3	OC3PEN	R/W	Output Compare Channel3 Preload Enable
6:4	OC3MOD	R/W	Output Compare Channel3 Mode Configure
7	OC3CEN	R/W	Output Compare Channel3 Clear Enable 0: OC3REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Selection This bit defines the input/output direction and the selected input pin. 00: CC4 channel is output 01: CC4 channel is input, and IC4 is mapped on TI4 10: CC4 channel is input, and IC4 is mapped on TI3 11: CC4 channel is input, and IC4 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC4EN=0).
10	OC4FEN	R/W	Output Compare Channel4 Preload Enable
11	OC4PEN	R/W	Output Compare Channel4 Buffer Enable
14:12	OC4MOD	R/W	Output Compare Channel4 Mode Configure
15	OC4CEN	R/W	Output Compare Channel4 Clear Enable

Input capture mode:

Field	Name	R/W	Description
1:0	CC3SEL	R/W	Capture/Compare Channel 3 Select 00: CC3 channel is output 01: CC3 channel is input, and IC3 is mapped on TI3 10: CC3 channel is input, and IC3 is mapped on TI4 11: CC3 channel is input, and IC3 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC3EN=0).
3:2	IC3PSC	R/W	Input Capture Channel 3 Perscaler Configuration 00：PSC=1 01：PSC=2 10：PSC=4 11：PSC=8

Field	Name	R/W	Description
			PSC is prescaled factor, which triggers capture once every PSC events.
7:4	IC3F	R/W	Input Capture Channel 3 Filter Configuration
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Select 00: CC4 channel is output 01: CC4 channel is input, and IC4 is mapped on TI4 10: CC4 channel is input, and IC4 is mapped on TI3 11: CC4 channel is input, and IC4 is mapped on TRC, and only works in internal trigger input Note: This bit can be written only when the channel is disabled (TMRx_CCEN register CC4EN=0).
11:10	IC4PSC	R/W	Input Capture Channel 4 Perscaler Configuration
15:12	IC4F	R/W	Input Capture Channel 4 Filter Configuration

16.6.9 Enable capture/compare channel register (TMRx_CCEN)

Offset address: 0x20

Reset value: 0x0000

Field	Name	R/W	Description
0	CC1EN	R/W	Capture/Compare Channel1 Output Enable **When the capture/compare channel 1 is configured as output:** 0: Output is disabled 1: Output is enabled **When the capture/compare channel 1 is configured as input:** This bit determines whether the value CNT of the counter can be captured and enter TMRx_CC1 register 0: Capture is disabled 1: Capture is enabled
1	CC1POL	R/W	Capture/Compare Channel1 Output Polarity Configure **When CC1 channel is configured as output:** 0: OC1 high level is valid 1: OC1 low level is valid **When CC1 channel is configured as input:** 0: Phase not reversed: capture at the rising edge of IC1; phase not reversed when IC1 is used as external trigger. 1: Phase reversed, capature at the falling edge of ICC1; phase reversed when IC1 is used as external trigger. Note: When the protection level is 2 or 3, this bit cannot be modified
3:2	Reserved		
4	CC2EN	R/W	Capture/Compare Channel2 Output Enable Refer to CCEN_CC1EN
5	CC2POL	R/W	Capture/Compare Channel2 Output Polarity Configure Refer to CCEN_CC1POL
7:6	Reserved		
8	CC3EN	R/W	Capture/Compare Channel3 Output Enable Refer to CCEN_CC1EN

Field	Name	R/W	Description
9	CC3POL	R/W	Capture/Compare Channel3 Output Polarity Configure Refer to CCEN_CC1POL
11:10	Reserved		
12	CC4EN	R/W	Capture/Compare Channel4 Output Enable Refer to CCEN_CC1EN
13	CC4POL	R/W	Capture/Compare Channel4 Output Polarity Refer to CCEN_CC1POL
15:14	Reserved		

Table 75Output Control Bit of Standard OCx Channel

CCxEN bit	OCx output state
0	Output is disabled (OCx=0, OCx_EN=0)
1	OCx=OCxREF+polarity, OCx_EN=1

Note: The state of external I/O pin connected to the standard OCx channel depends on the state of the OCx channel and the GPIO and AFIO registers.

16.6.10 Counter register (TMRx_CNT)

Offset address: 0x24

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CNT	R/W	Counter Value

16.6.11 Prescaler register (TMRx_PSC)

Offset address: 0x28

Reset value: 0x0000

Field	Name	R/W	Description
15:0	PSC	R/W	Prescaler Value Clock frequency of counter (CK_CNT)＝f_{CK_PSC}/(PSC+1)

16.6.12 Auto reload register (TMRx_AUTORLD)

Offset address: 0x2C

Reset value: 0xFFFF

Field	Name	R/W	Description
15:0	AUTORLD	R/W	Auto Reload Value When the value of auto reload is empty, the counter will not count.

16.6.13 Channel 1 capture/compare register (TMRx_CC1)

Offset address: 0x34

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC1	R/W	Capture/Compare Channel 1 Value **When the capture/compare channel 1 is configured as input mode:** CC1 contains the counter value transmitted by the last input capture channel 1 event. **When the capture/compare channel 1 is configured as output mode:** CC1 contains the current load capture/compare register value Compare the value CC1 of the capture/compare channel 1 with the value CNT of the counter to generate the output signal on OC1. When the output compare preload is disabled (OC1PEN=0 for TMRx_CCM1 register), the written value will immediately affect the output compare results; If the output compare preload is enabled (OC1PEN=1 for TMRx_CCM1 register), the written value will affect the output compare result when an update event is generated.

16.6.14 Channel 2 capture/compare register (TMRx_CC2)

Offset address: 0x38

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC2	R/W	Capture/Compare Channel 2 Value Refer to TMRx_CC1

16.6.15 Channel 3 capture/compare register (TMRx_CC3)

Offset address: 0x3C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC3	R/W	Capture/Compare Channel 3 Value Refer to TMRx_CC1

16.6.16 Channel 4 capture/compare register (TMRx_CC4)

Offset address: 0x40

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC4	R/W	Capture/Compare Channel 4 Value Refer to TMRx_CC1

16.6.17 DMA control software (TMRx_DCTRL)

Offset address: 0x48

Reset value: 0x0000

Field	Name	R/W	Description
4:0	DBADDR	R/W	DMA Base Address Setup These bits define the base address of DMA in continuous mode (when reading or writing TMRx_DMADDR register), and DBADDR is defined as the offset from the address of TMRx_CTRL1 register: 00000: TMRx_CTRL1 00001: TMRx_CTRL2

Field	Name	R/W	Description
			00010: TMRx_SMCTRL …….
7:5	Reserved		
12:8	DBLEN	R/W	DMA Burst Transfer Length Setup These bits define the transfer length and transfer times of DMA in continuous mode. The data transferred can be 16 bits and 8 bits. When reading/writing TMRx_DMADDR register, the timer will conduct a continuous transmission; 00000: Transmission once 00001: Transmission twice 00010: Transmission for three times …… 10001: Transmission for 18 times The transmission address formula is as follows: Transmission address=TMRx_CTRL1 address (slave address) +DBADDR+DMA index; DMA index=DBLEN For example: DBLEN=7, DBADDR=TMR1_CTRL1 (slave address) means the address of the data to be transmitted, while the address +DBADDR+7 of TMRx_CTRL1 means the address of the data to be written/read, Data transmission will occur to: TMRx_CTRL1 address + seven registers starting from DBADDR. The data transmission will change according to different DMA data length: 1) When the transmission data is set to 16 bits, the data will be transmitted to seven registers 2) When the transmission data is set to 8 bits, the data of the first register is the MSB bit of the first data, the data of the second register is the LSB bit of the first data, and the data will still be transmitted to seven registers.
15:13	Reserved		

16.6.18 DMA address register of continuous mode (TMRx_DMADDR)

Offset address: 0x4C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	DMADDR	R/W	DMA Register for Burst Transfer Read or write operation access of TMRx_DMADDR register may lead to access operation of the register in the following address: TMRx_CTRL1 address + (DBADDR+DMA index) ×4 Wherein: "TMRx_CTRL1 address" is the address of control register 1 (TMRx_CTRL1); "DBADDR" is the base address defined in TMRx_DCTRL register; "DMA index" is the offset automatically controlled by DMA, and it depends on DBLEN defined in TMRx_DCTRL register.

17 Basic Timer (TMR6/7)

17.1 Introduction

The basic timers TMR6 and TMR7 have an unsigned 16-bit counter, auto reload register, prescaler and trigger controller.

The basic timer provides time reference for general-purpose timer and provides clock for DAC. DMA request can be generated by configuration.

17.2 Main Characteristics

（1） Counter: 16-bit counter, which can only count up

（2） Prescaler: 16-bit programmable prescaler

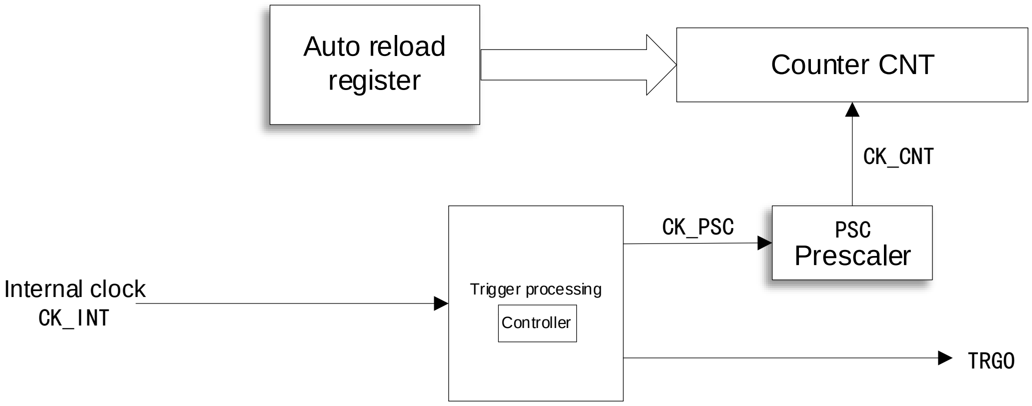
（3） Clock source: There is only internal clock

（4） Single-pulse mode

（5） Provide clock for DAC

17.3 Structure Block Diagram

Figure 61 Basic Timer Structure Block Diagram



17.4 Functional Description

17.4.1 Clock Source Selection

The basic timer is driven by internal clock source TMRx_CLK

Configure the CNTEN bit of TMRx_CTRL1 register to enable the counter; when CNTEN bit is set, the internal clock CK_INT can generate CK_INT to drive the counter through the controller and prescaler.

17.4.2 Timebase Unit

The time base unit in the basic timer contains three registers:

- Counter register (CNT) 16 bits
- Auto reload register (AUTORLD) 16 bits
- Prescaler (PSC) 16 bits

Counter CNT

The basic timer only has one count mode: count-up

Count-up mode

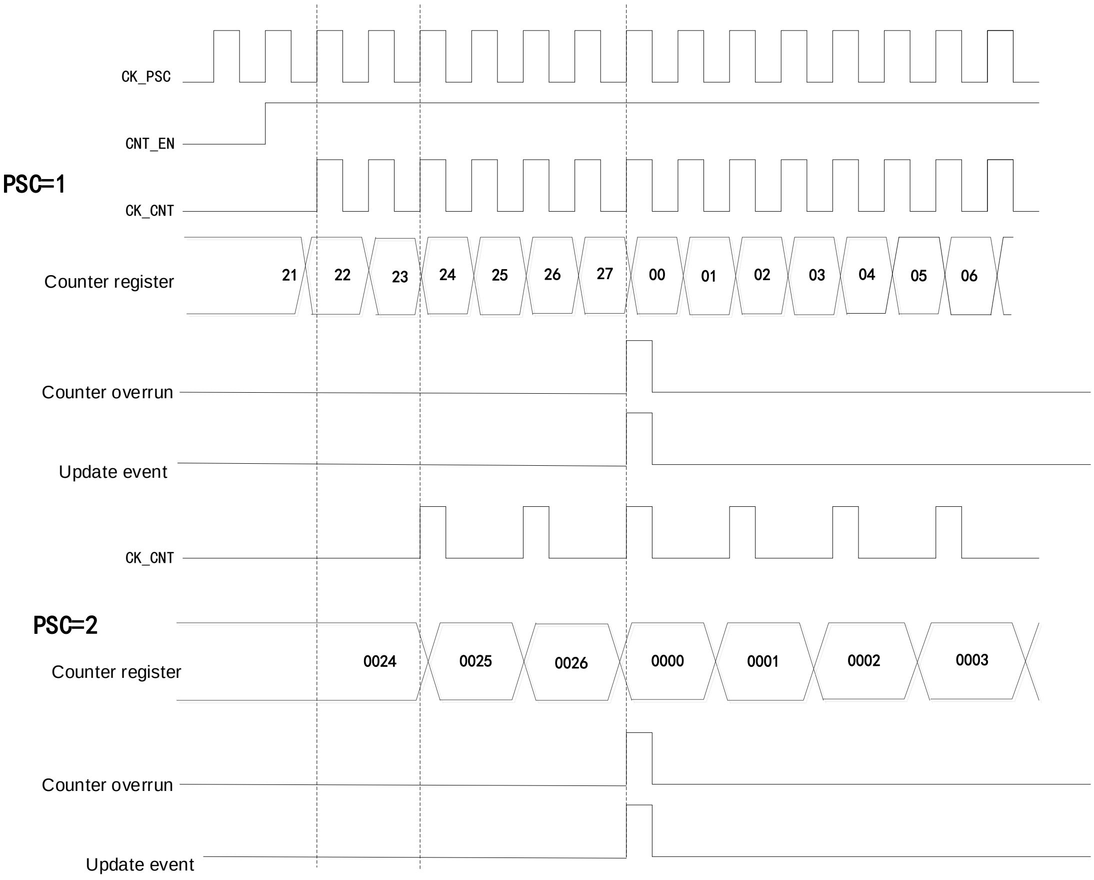
When the counter is in count-up mode, the counter will count up from 0; every time a pulse is generated, the counter will increase by 1 and when the value of the counter (TMRx_CNT) is equal to the value of the auto reload (TMRx_AUTORLD), then the counter will start to count again from 0, a count-up overrun event will be generated, and the value of the auto reload (TMRx_AUTORLD) is written in advance.

Disable the update event and set UD bit of TMRx_CTRL1 register to 1.

Generate the update interrupt or DMA request and set URSSEL bit in TMRx_CTRL1 register.

When an update event occurs, both the auto reload register and the prescaler register will be updated.

Figure 62 Timer Timing Diagram, the internal clock division factor is 1 or 2



Prescaler PSC

The prescaler is 16 bits and programmable, and it can divide the clock frequency of the counter to any value between 1 and 65536 (controlled by TMRx_PSC register), and after frequency division, the clock will drive the counter CNT to count. The prescaler has a buffer, which can be changed during running.

17.5 Register Address Mapping

In the following table, all registers of the basic timer are mapped to a 16-bit addressable (address) space.

Table 76 Basic Timer Register Mapping

Register name	Description	Offset address
TMRx_CTRL1	Control register 1	0x00
TMRx_CTRL2	Control register 2	0x04
TMRx_DIEN	DMA/Interrupt enable register	0x0C
TMRx_STS	State register	0x10

Register name	Description	Offset address
TMRx_CEG	Control event generation register	0x14
TMRx_CNT	Counter register	0x24
TMRx_PSC	Prescaler register	0x28
TMRx_AUTORLD	Auto reload register	0x2C

17.6 Register Functional Description

17.6.1 Control register 1 (TMRx_CTRL1)

Offset address: 0x00

Reset value: 0x0000

Field	Name	R/W	Description
0	CNTEN	R/W	Counter Enable 0: Disable 1: Enable When the timer is configured as external clock, gated mode and encoder mode, it is required to write 1 to the bit by software to start regular work; when it is configured as the trigger mode, it can be written to 1 by hardware.
1	UD	R/W	Update Disable Update event can cause AUTORLD, PSC and CCx to generate the value of update setting. 0: Update event is allowed (UEV) An update event can occur in any of the following situations: The counter overruns/underruns; Set UEG bit; Update generated by slave mode controller. 1: Update event is disabled
2	URSSEL	R/W	Update Request Source Select If interrupt or DMA is enabled, the update event can generate update interrupt or DMA request. Different update request sources can be selected through this bit. 0: The counter overruns or underruns Set UEG bit Update generated by slave mode controller 1: The counter overruns or underruns
3	SPMEN	R/W	Single Pulse Mode Enable When an update event is generated, the output level of the channel can be changed; in this mode, the CNTEN bit will be cleared, the counter will be stopped, and the output level of the channel will not be changed. 0: Disable 1: Enable
6:4	Reserved		

Field	Name	R/W	Description
7	ARPEN	R/W	Auto-reload Preload Enable When the buffer is disabled, the program modification TMRx_AUTORLD will immediately modify the values loaded to the counter; when the buffer is enabled, the program modification TMRx_AUTORLD will modify the values loaded to the counter in the next update event. 0: Disable 1: Enable
15:8	Reserved		

17.6.2 Control register 2 (TMRx_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
2:0	Reserved		
3	CCDSEL	R/W	Capture/compare DMA Select 0: Send DMA request of CCx when CCx event occurs 1: Send DMA request of CCx when an update event occurs
6:4	MMSEL	R/W	Master Mode Signal Select The signals of timers working in master mode can be used for TRGO, which affects the work of timers in slave mode and cascaded with master timer, and specifically affects the configuration of timers in slave mode. 000: Reset; the reset signal of master mode timer is used for TRGO 001: Enable; the counter enable signal of master mode timer is used for TRGO 010: Update; the update event of master mode timer is used for TRGO 011: Compare pulses; when the master mode timer captures/compares successfully (CCxIFLG=1), a pulse signal is output for TRGO 100: Compare mode 1; OC1REF is used to trigger TRGO 101: Compare mode 2; OC2REF is used to trigger TRGO 110: Compare mode 3; OC3REF is used to trigger TRGO 111: Compare mode 4; OC4REF is used to trigger TRGO
15:7	Reserved		

17.6.3 DMA/Interrupt enable register (TMRx_DIEN)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
0	UIEN	R/W	Update interrupt Enable 0: Disable 1: Enable
7:1	Reserved		
8	UDIEN	R/W	Update DMA Request Enable 0: Disable 1: Enable
15:9	Reserved		

17.6.4 State register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
0	UIFLG	RC_W0	Update Event Interrupt Generate Flag 0: Update event interrupt does not occur 1: Update event interrupt occurs When the counter value is reloaded or reinitialized, an update event will be generated. The bit is set to 1 by hardware and cleared by software; update events are generated in the following situations: (1) UD=0 on TMRx_CTRL1 register, and when the value of the repeat counter overruns/underruns, an update event will be generated; (2) URSEL=0 and UD=0 on TMRx_CTRL1 register, configure UG = 1 on TMRx_CEG register to generate update event, and the counter needs to be initialized by software; (3) URSEL=0 and UD=0 on TMRx_CTRL1 register, generate update event when the counter is initialized by trigger event.
15:1	Reserved		

17.6.5 Control event generation register (TMRx_CEG)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
0	UEG	W	Update Event Generate 0: Invalid 1: Initialize the counter and generate the update event This bit is set to 1 by software, and cleared by hardware. Note: When an update event is generated, the counter of the prescaler will be cleared, but the prescaler factor remains unchanged. In the count-down mode, the counter reads the value of TMRx_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared.
15:1	Reserved		

Note: The state of external I/O pin connected to the standard OCx channel depends on the state of the OCx channel and the GPIO and AFIO registers.

17.6.6 Counter register (TMRx_CNT)

Offset address: 0x24

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CNT	R/W	Counter Value

17.6.7 Prescaler register (TMRx_PSC)

Offset address: 0x28

Reset value: 0x0000

Field	Name	R/W	Description
15:0	PSC	R/W	Prescaler Value Clock frequency of counter (CK_CNT)＝f_{CK_PSC}/(PSC+1)

17.6.8 Auto reload register (TMRx_AUTORLD)

Offset address: 0x2C

Reset value: 0xFFFF

Field	Name	R/W	Description
15:0	AUTORLD	R/W	Auto Reload Value When the value of auto reload is empty, the counter will not count.

18 Watchdog Timer (WDT)

18.1 Introduction

The watchdog is used to monitor system failures caused by software errors. There are two watchdog devices on the chip: independent watchdog and window watchdog, which improve the security, and make the time more accurate and the use more flexible.

The independent watchdog will reset only when the counter is reduced to 0, and the value of refresh counter will not be reset until it is not reduced to 0.

The window watchdog will reset when the counter decreases to 0x3F. When the count value of the counter is before the window value of the configuration register, the refresh counter will also be reset.

18.2 Independent Watchdog Timer (IWDT)

18.2.1 Introduction

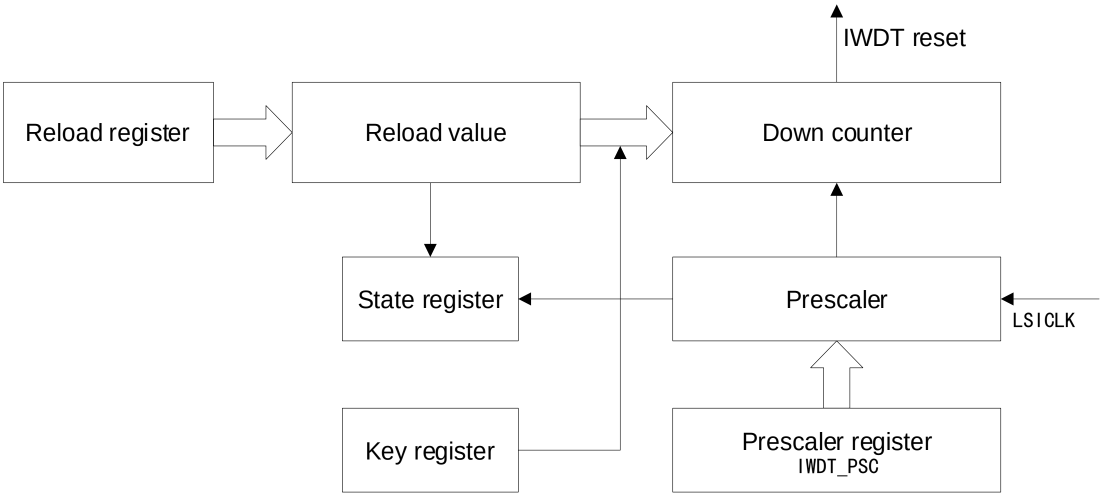
The independent watchdog consists of an 8-bit prescaler IWDT_PSC, 12-bit count-down counter, 12-bit reload register IWDT_CNTRLD, key register IWDT_KEY and state register IWDT_STS.

The independent watchdog has an independent clock source, and even if the master clock fails, it is still valid.

The independent watchdog is applicable to the situations where an independent environment is required but the accuracy requirement is not high.

18.2.2 Structure Block Diagram

Figure 63 Independent Watchdog Structure Block Diagram



Note: The watchdog function is in the V_{DD} power supply area and can work normally in the shutdown or standby mode.

18.2.3 Functional Description

18.2.3.1 Key register

Write 0xCCCC in the key register to enable the independent watchdog, then the counter starts to count down, and when the counter counts to 0x000, a reset will be generated.

Write 0xAAAA in the key register, and the value of the reload register will be reloaded to the counter to prevent the watchdog from resetting.

Write 0X5555 in the key register to rewrite the value of the prescaler register and the reload register.

18.2.3.2 Regiser access protection

The prescaler register and reload register have the function of write protection. If you want to rewrite these two registers, you need to write 0X5555 in the key register. If you write other value in the key register, the protection of the register will be started again.

Write 0xAAAA to the key register and the write protection function will also be enabled.

18.2.3.3 Hardware watchdog

After the "hardware watchdog" function is enabled, and the system is powered on and reset, the watchdog will run automatically. If 0xAAAA is not written to the key register, reset will be generated after the counter finishes counting.

18.2.3.4 Debug mode

The independent watchdog can be configured in debug mode and choose to stop or continue to work. Depend on DBGMCU_CFG register IWDT_STS bit.

18.3 Window Watchdog Timer (WWDT)

18.3.1 Introduction

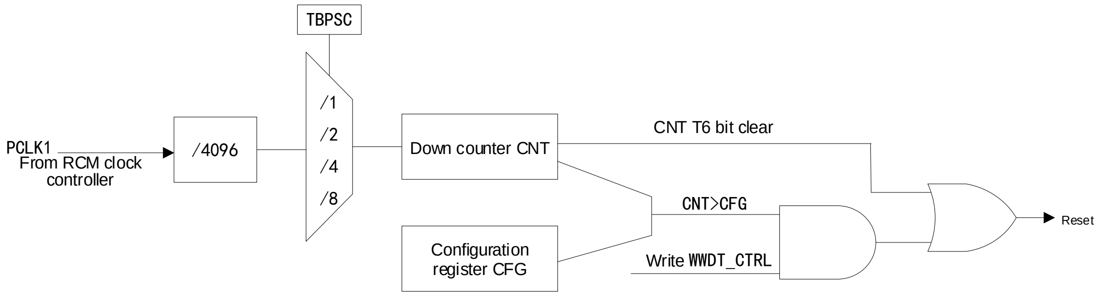
The window watchdog contains a 7-bit free-running down counter, prescaler and control register WWDT_CTRL, configuration register WWDT_CFG and state register WWDT_STS.

The window watchdog clock comes from PCLK1, and the counter clock is obtained from the CK counter clock through frequency division by prescaler (configured by the configuration register).

The window watchdog is applicable when precise timing is needed.

18.3.2 Structure Block Diagram

Figure 64 Window Watchdog Structure Block Diagram



18.3.3 Functional Description

Enable window watchdog timer; the reset conditions are:

- When the counter count is less than 0x40, a reset will be generated.
- The reload counter will be reset before the counter counts to the value of the window register.

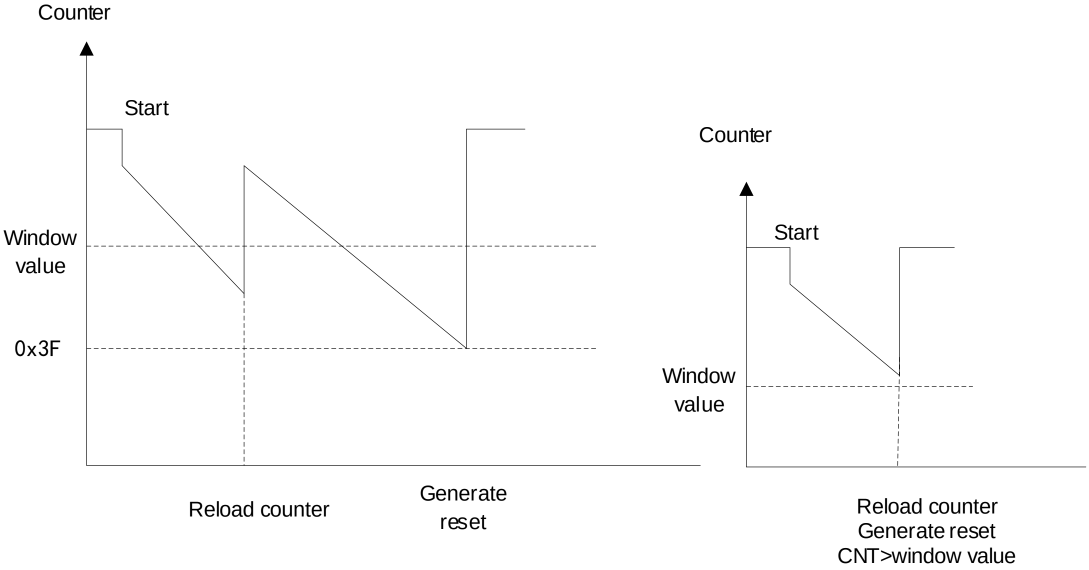
After reset, the watchdog is always closed and the watchdog can be enabled only by setting the WWDTEN bit of WWDT_CTRL control register.

The counter of window watchdog is in free state. When the watchdog is disabled, the counter will continue to count down. The counter must be reloaded between the value of window register and 0x40 to avoid reset.

Setting the EWIEN bit of the configuration register can enable the early wake-up interrupt. When the count reaches 0x40, the interrupt will be generated. Entering the interrupt service program (ISTS) can be used to prevent the window watchdog from resetting. EWIEN interrupt can be cleared by writing 0 in the state register.

The unique window of the window watchdog timer can effectively monitor whether the program is faulty. For example, assuming that the running time of a program segment is T, and the value of the window register is slightly less than (TR-T), if there is no reload register in the window, it means that the program is faulty, and when the counter counts to 0x3F, it will generate reset.

Figure 65 Window Watchdog Timing Diagram



The calculation formula of window watchdog timer timeout is as follows:

$$T_{WWDT}=T_{PCLK1}\times 2^{WTB}\times（T[5:0]+1）$$

Wherein:

- T_{WWDT}: WWDT timeout
- T_{PCLK1}: Clock cycle of APB1 in ms

Minimum/Maximum timeout when PCLK1=36MHZ

WTB	Minimum timeout value	Maximum timeout value
0	113μs	7.28ms
1	227μs	14.56ms
2	455μs	29.12ms
3	910μs	58.25ms

18.4 IWDT Register Address Mapping

Table 77 IWDT Register Mapping

Register name	Description	Offset address
IWDT_KEY	Key register	0x00
IWDT_PSC	Prescaler register	0x04
IWDT_CNTRLD	Counter reload register	0x08
IWDT_STS	State register	0x0C

18.5 IWDT Register Functional Description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

18.5.1 Key register (IWDT_KEY)

Offset address: 0x00

Reset value: 0x0000 0000 (reset in standby mode)

Field	Name	R/W	Description
15:0	KEY	W	Allow Access IWDT Register Key Value Writing 0x5555 means enabled access to IWDT_PSC and IWDT_CNTRLD registers; When the software writes 0xAAAA, it means to execute the reload counter, and a certain interval is required to prevent the watchdog from resetting. Write 0xCCCC and the watchdog will be enabled (the hardware watchdog is unrestricted by this command word); This register is write-only and the read-out vlue is 0x0000.
31:16	Reserved		

18.5.2 Prescaler register (IWDT_PSC)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
2:0	PSC	R/W	Prescaler Factor Configure Support write protection function; when writing 0x5555 in the IWDT_KEY register, it is allowed to access the register; in the process of writing this register, only when IWDT_STS register PSCUFLG=0, can the prescaler factor be changed; in the process of reading this register, only when PSCUFLG=0, can the read-out value of PSC register be valid. 000: PSC=4 001: PSC=8 010: PSC=16 011: PSC=32 100: PSC=64 101: PSC=128 110: PSC=256 111: PSC=256
31:3	Reserved		

18.5.3 Counter reload register (IWDT_CNTRLD)

Offset address: 0x08

Reset value: 0x0000 0FFF (reset in standby mode)

Field	Name	R/W	Description
11:0	CNTRLD	R/W	Watchdog Counter Reload Value Setup It supports write protection function and defines the value loaded to the watchdog counter when 0xAAAA is written by IWDT_KEY register; in the process of writing this register, this register can be modified only when

Field	Name	R/W	Description
			CNTUFLG=0. In the process of reading this register, when CNTUFLG=0 in IWDT_STS register, the read value is valid. The watchdog timeout cyclecan be calculated by the reload value and clock prescaled value.
31:12	Reserved		

18.5.4 State register (IWDT_STS)

Offset address: 0x0C

Reset value: 0x0000 0000 (not reset in standby mode)

Field	Name	R/W	Description
0	PSCUFLG	R	Watchdog Prescaler Factor Update Flag When the prescaler factor is updated, it is set to 1 by hardware; after the prescaler factor is updated, the bit is cleared by hardware; the prescaler factor is updated only when the PSCUFLG bit is cleared.
1	CNTUFLG	R	Watchdog Counter Reload Value Update Flag When the counter reload value is updated, it is set to 1 by hardware; after the counter reload value is updated, the bit is cleared by hardware; the counter reload value is updated only when the CNTUFLG bit is cleared.
31:2	Reserved		

18.6 WWDT Register Address Mapping

Table 78 WWDT Register Mapping

Register name	Description	Offset address
WWDT_CTRL	Control register	0x00
WWDT_CFG	Configuration register	0x04
WWDT_STS	State register	0x08

18.7 WWDT Register Functional Description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

18.7.1.1 Control register (WWDT_CTRL)

Offset address: 0x00

Reset value: 0x0000 007F

Field	Name	R/W	Description
6:0	CNT	R/W	Counter Value Setup This counter is 7 bits, and CNT6 is the most significant bit These bits are used to store the counter value of the watchdog. When the count value decreases from 0x40 to 0x3F, WWDT reset will be generated.

Field	Name	R/W	Description
7	WWDTEN	R/S	Window Watchdog Enable This bit is set to 1 by software and can be cleared by hardware only after reset. When WWDTEN=1, WWDT can generate a reset. 0: Disable 1: Enable
31:8	Reserved		

18.7.1.2 Configuration register (WWDT_CFG)

Offset address: 0x04

Reset value: 0x0000 007F

Field	Name	R/W	Description
6:0	WIN	R/W	Window Value Setup This window value is 7 bits, which is used to compare with the down counter.
8:7	TBPSC	R/W	Timer Base Prescaler Factor Configure Divide the frequency on the basis of PCLK1/4096 00: No frequency division 01: Two-divided frequency 10: Four-divided frequency 11: Eight-divided frequency
9	EWIEN	R/S	Early Wakeup Interrupt Enable 0: No effect 1: When the counter value reaches 0x40, an interrupt will be generated; this interrupt is cleared by hardware after reset.
31:10	Reserved		

18.7.1.3 State register (WWDT_STS)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	EWIFLG	RC_W0	Early Wakeup Interrupt Occur Flag 0: Not occur 1: When the counter value reaches 0x40, it is set to 1 by hardware; if the interrupt is not enabled, the bit will also be set to 1; it can be cleared by writing 0 by software.
31:1	Reserved		

19 Real-time Clock (RTC)

19.1 Full Name and Abbreviation Description of Terms

Table 79 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Second	SEC
Alarm	ALR
Overflow	OVR
Prescaler	PSC
Time Basic Clock	TBCLK

19.2 Main Characteristics

Real-time clock (RTC) is a timer that automatically switches to backup power supply after main power failure to maintain the operation.

（1） Timebase unit

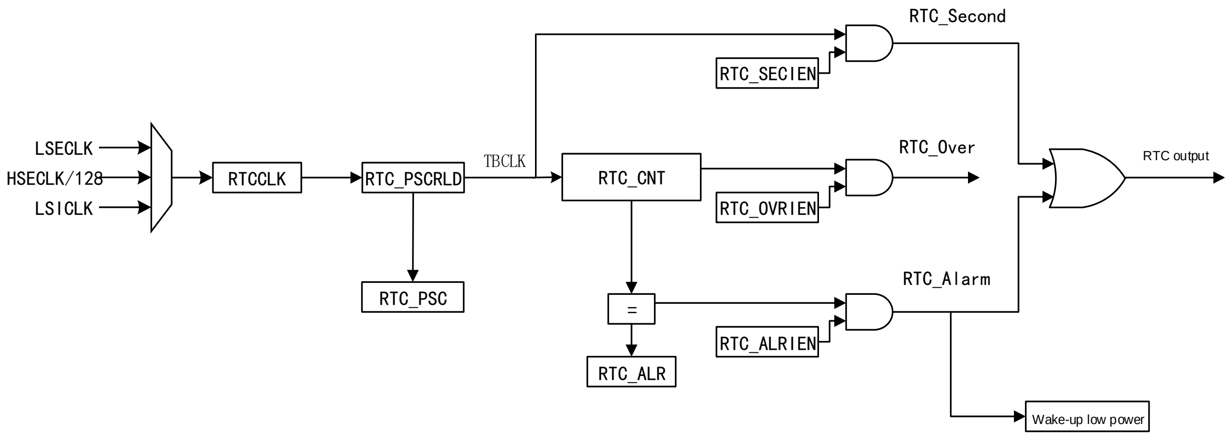
（2） Programmable 32bit counter

（3） Multiple interrupt control

（4） Automatic wakeup of low power

19.3 Structure Block Diagram

Figure 66 RTC Structure Block Diagram



19.4 Functional Description

19.4.1 Timebase Unit

Clock source

RTC has three clock sources:

- External LSECLK crystal oscillator
- External HSECLK crystal oscillator 128 divided frequency
- Internal LSICLK

Users could select clock source by configuring registers of Clock Control Management（CCM）.

Prescaler

The RTC prescaler contains a 20-bit programmable frequency divider, which can be programmed to generate RTC time reference of up to 1 second.

19.4.2 RTC Register Configuration

In order to prevent counting exception caused by accidental write in RTC register, RTC adopts write protection mechanism. Only when the write protection is removed, can the register with write protection function be operated.

When configuring RTC clock, it's required to set BPWEN bit of the power control register (PMU_CTRL) to "1"; configure CFGMFLG bit of RTC_CSTS register to make the RTC enter the configuration mode so that the RTC_PSCRLD, RTC_CNT and RTC_ALR registers can be configured; clear CFGMFLG bit of RTC_CSTS register to exit the configuration mode.

The write operation to any register of RTC can be performed only after the previous write operation is finished (judge by querying RTC_CSTS OCFLG).

19.4.3 Programmable Alarm

As a real-time clock, RTC integrates the alarm function internally, and it runs mainly through alarm register and counter, and configures the alarm time through register RTC_ ALR; after the alarm function is enabled, when the counter value is equal to the alarm value, it will be triggered and the alarm flag will be set. If the alarm interrupt is enabled, the interrupt processing will be triggered, and through the configuration of external line 17 interrupt, RTC alarm can be used to wake up low power consumption.

19.4.4 RTC Output

RTC can output the internal RTC second pulse, alarm signal and calibration clock to the outside through PC13 pin, and select the output pulse by configuring BAKPR_CLKCAL register.

19.4.5 Interrupt

RTC can generate second interrupt, alarm interrupt and overrun interrupt. When 20-bit prescaler overrun, alarm event and 32-bit counter overrun are generated, the corresponding state flag bit will be pending and the corresponding interrupt can be generated by configuring RTC_CTRL register.

19.5 Register Address Mapping

Table 80 RTC Register Address Mapping

Register name	Description	Offset address
RTC_CTRL	RTC control register	0x00
RTC_CSTS	RTC control/state register	0x04
RTC_PSCRLDH	RTC prescaler reload register high bit	0x08
RTC_PSCRLDL	RTC prescaler reload register low bit	0x0C
RTC_PSCH	RTC prescaler register high bit	0x10
RTC_PSCL	RTC prescaler register low bit	0x14
RTC_CNTH	RTC counter register high bit	0x18
RTC_CNTL	RTC counter register low bit	0x1C
RTC_ALRH	RTC alarm value register high bit	0x20
RTC_ALRL	RTC alarm value register low bit	0x24

19.6 Register Functional Description

19.6.1 RTC control register (RTC_CTRL)

Offset address: 0x00

Reset value: 0x0000

Field	Name	R/W	Description
0	SECIEN	R/W	Second Interrupt Enable 0: Disable 1: Enable
1	ALRIEN	R/W	Alarm Interrupt Enable 0: Disable 1: Enable
2	OVRIEN	R/W	Overflow Interrupt Enable 0: Disable 1: Enable
15:3	Reserved		

19.6.2 RTC control/state register (RTC_CSTS)

Offset address: 0x04

Reset value: 0x0020

Field	Name	R/W	Description
0	SECFLG	RC_W0	Second Signal Condition Met Flag This flag can provide a periodic signal (usually 1 second) for the RTC counter. When the 32-bit programmable prescaler overruns, it is set to 1 by hardware and the RTC counter will add by 1; it can be only cleared by writing 0 by software. 0: No second flag 1: Second flag
1	ALRFLG	RC_W0	Alarm Occur Flag When the counter reaches RTC_ALR value, it is set to 1 by hardware; it can be only cleared by writing 0 by software. 0: No alarm 1: Alarm
2	OVRFLG	RC_W0	Overflow Occur Flag When the counter overruns, it is set to 1 by hardware; it can be only cleared by writing 0 by software. 0: No overrun 1: 32-bit programmable counter overrun
3	RSYNCFLG	RC_W0	Registers Synchronized Flag When RTC_CNT, RTC_PSCRLD and RTC_ALR registers have been synchronized, it is set to 1 by hardware; it can be cleared by writing 0 by software. After the APB1 clock is reset or stopped, this bit must be cleared by software, and the user program can correctly read out the values of RTC_CNT, RTC_PSCRLD and RTC_ALR only when it is set to 1 by hardware. 0: Not synchronized 1: Synchronized
4	CFGMFLG	R/W	Configure Mode Enable Flag Write operation can be performed for RTC_CNT, RTC_ALR or RTC_PSCRLD registers only after writing 1 by software and entering the configuration mode; exit the configuration mode after writing 0 by software. 0: Exit configuration mode (start to update RTC register) 1: Enable configuration mode
5	OCFLG	R	RTC Operation Complete Flag Indicate the state of last write of RTC register. 0: Uncompleted; next write operation cannot be executed 1: Completed; next write operation can be executed
15:6	Reserved		

19.6.3 RTC prescaler reload register (RTC_PSCRLD)

This register saves the cycle count value of RTC prescaler, and only when the OCFLG value is 1, can the write operation be performed.

RTC prescaler load register high bit (RTC_PSCRLDH)

Offset address: 0x08

Reset value: 0x0000

Field	Name	R/W	Description
3:0	PSCRLDH[19:16]	W	RTC Prescaler Reload Value High Setup These bits can be used to define the frequency of time base clock according to the following formula: f_{TBCLK}=f_{RTCCLK}/(RLD [19:0]+1)
15:4	Reserved		

RTC prescaler load register low bit (RTC_PSCRLDL)

Offset address: 0x0C

Reset value: 0x8000

Field	Name	R/W	Description
15:0	PSCRLDL[15:0]	W	RTC Prescaler Reload Value Low Setup These bits can be used to define the frequency of time base clock according to the following formula: f_{TBCLK}=f_{RTCCLK}/(RLD[19:0]+1)

Note: If the input clock frequency is 32.768kHz (f_{RTCCLK}), write 7FFFh in this register to obtain a signal with a cycle of 1 second.

19.6.4 RTC prescaer register (RTC_PSC)

This register saves the value of RTC_PSCRLD, which is read-only, and can be reloaded by hardware when RTC_PSCRLD or RTC_CNT register changes.

RTC prescaler register high bit (RTC_PSCH)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
3:0	PSCH[19:16]	R	RTC Clock Prescaler High Setup
15:4	Reserved		

RTC prescaler register low bit (RTC_PSCL)

Offset address: 0x14

Reset value: 0x8000

Field	Name	R/W	Description
15:0	PSCL[15:0]	R	RTC Clock Prescaler Low Setup

19.6.5 RTC counter register (RTC_CNT)

When the OCFLG value is 1, write operation is allowed; when read operation is performed, the count value (system time) in the counter will be returned directly.

RTC counter register high bit (RTC_CNTH)

Offset address: 0x18

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CNTH[31:16]	R/W	RTC Counter High Setup

RTC counter register low bit (RTC_CNTL)

Offset address: 0x1C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CNTL[15:0]	R/W	RTC Counter Low Setup

19.6.6 RTC alarm value register (RTC_ALR)

Write operation can be performed when OCFLG value is 1.

RTC alarm value register high bit (RTC_ALRH)

Offset address: 0x20

Reset value: 0xFFFF

Field	Name	R/W	Description
15:0	ALRH[31:16]	W	RTC Alarm Value High Setup

RTC alarm value register low bit (RTC_ALRL)

Offset address: 0x24

Reset value: 0xFFFF

Field	Name	R/W	Description
15:0	ALRL[15:0]	W	RTC Alarm Value Low Setup

20 Universal Synchronous/Asynchronous Transceiver (USART)

20.1 Full Name and Abbreviation Description of Terms

Table 81 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Clear to Send	CTS
Request to Send	RTS
Most Significant Bit	MSB
Least Significant Bit	LSB
Guard	GRD
Overrun	OVR

20.2 Introduction

USART (universal synchronous/asynchronous transceiver) is a serial communication device that can flexibly exchange full-duplex and half-duplex data with external devices, and meets the requirements of external devices for industry standard NRZ asynchronous serial data format. USART also provides a wide range of baud rate for selection and supports multiprocessor communication.

USART not only supports standard asynchronous transceiver mode, but also supports some other serial data exchange modes, such as LIN protocol, smart card protocol, IrDA SIR ENDEC specification and hardware flow control mode.

USART also supports DMA function to realize high-speed data communication.

20.3 Main Characteristics

（1） Full duplex asynchronous communication

（2） Single-line half-duplex communication

（3） NRZ standard format

（4） Characteristics of programmable serial port:

- Data bit: 8 or 9 bits
- Check bits: Even parity check, odd parity check, no check
- Support 0.5, 1, 1.5 and 2 stop bits

（5） Check control

- Transmit the check bit
- Check the received data

（6） Independent transmitter and receiver enable bit

（7） programmable baud rate generator, with baud rate of up to 4.5Mbits/s

（8） Multiprocessor communication:

- If the address does not match, it will enter the mute mode
- Wake up from mute mode through idle bus detection or address flag detection

（9） Synchronous transmission mode

（10） Generation and detection of LIN break frame

（11） Support the smart card interface of ISO7816-3 standard

（12） Support IrDA protocol

（13） Support hardware flow control

（14） DMA can be used for continuous communication

（15） State flag bit:

- Transmission detection flag: The transmit register is empty, the receive register is not empty, and transmission is completed
- Error detection flag: Overrun error, noise error, parity error, frame error

（16） Multiple interrupt sources:

- The transmit register is empty
- Transmission is completed
- CTS changed
- The receive register cannot be empty
- Overload error
- Bus idle
- Parity error
- LIN disconnection detection
- Noise error
- Overrun error
- Frame error

20.4 Functional Description

Table 82 USART Pin Description

Pin	Type	Description
USART_RX	Input	Data receiving

Pin	Type	Description
USART_TX	Output I/O (single-line mode/smart card mode)	Data transmission When the transmitter is enabled and does not transmit data, the default is high
USART_CK	Output	Clock output
USART_nRTS	Input	Request to send in hardware flow control mode
USART_nCTS	Output	Clear to send in hardware flow control mode
IrDA_RDI	Input	Data input in IrDA mode
IrDA_TDO	Output	Data output in IrDA mode

20.4.1 Single-line Half-duplex Communication

HDEN bit of USART_CTRL3 register determines whether to enter the single-line half-duplex mode.

When USART enters single-line half-duplex mode:

- The CLKEN and LINMEN bit of USART_CTRL2 register, and IREN and SCEN bits of USART_CTRL3 register must be cleared.
- RX pin is disabled.
- TX pin should be configured as open-drain output and connected with RX pin inside the chip.
- Sending data and receiving data can not be carried out at the same time. The data cannot be received before they are transmitted. If needing to receive data, enabling receiving can be turned on only after TXCFLG bit of USART_STS register is set to 1.
- If there is data conflict on the bus, software management is needed to allocate the communication process.

20.4.2 Frame Format

The frame format of data frame is controlled by USART_CTRL1 register

- DBLCFG bit controls the character length, which can be set to 8 or 9 bits.
- PCEN bit controls to enable the check bit or not.
- PCFG bit controls the parity bit to be odd or even.

Table 83 Frame Format

DBLCFG bit	PCEN bit	USART data frame
0	0	Start bit+8-bit data+stop bit
0	1	Start bit+7-bit data+odd-even parity check bit+stop bit
1	0	Start bit+9-bit data+stop bit
1	1	Start bit+8-bit data+odd-even parity check bit+stop bit

Configurable stop bit

Four different stop bits can be configured through STOPCFG bit of USART_CTRL2 register.

- 1 stop bit: Default stop bit.
- 0.5 stop bit: Used when receiving data in smart card mode.
- 2 stop bits: Used in normal mode, single-line mode and hardware flow control mode.
- 1.5 stop bits: Used when sending and receiving data in smart card mode.

Check bit

PCFG bit of USART_CTRL1 determines the parity check bit; when PCFG=0, it is even parity check, on the contrary, it is odd parity check.

- Even check: When the number of frame data and check bit 1 is even, the even check bit is 0; otherwise it is 1.
- Odd check: When the number of frame data and check bit 1 is even, the odd check bit is 1; otherwise it is 0.

20.4.3 Transmitter

When TXEN bit of the register USART_CTRL1 is set, the transmit shift register will output data through TX pin and the corresponding clock pulses will be output through CK pin.

20.4.3.1 Character Transmit

DuringTransmitting period of USART, the least significant bit of the data will be moved out by TX pin first. In this mode, USART_ DATA register has a buffer between the internal bus and the transmitter shift register.

A data frame is composed of the start bit, character and stop bit, so there is a low-level start bit in front of each character; then there is a high-level stop bits the number of which is configurable.

Transmission configuration steps

- Set UEN bit of USART_CTRL1 register to enable USART.
- Decide the word length by setting DBLCFG bit of USART_CTRL1 register.
- Decide the number of stop bits by setting STOPCFG bit of USART_CTRL2 register.
- If multi-buffer communication is selected, DMA should be enabled in USART_CTRL3 register.
- Set the baud rate of communication in USART_BR register.
- Enable TXEN bit in USART_CTRL1 register, and Transmit an idle frame.
- Wait for TXBEFLG bit of USART_STS register to be set to 1.

- Write data to USART_DATA register (if DMA is not enabled, repeat steps 7-8 for each byte to be transmitted).
- Wait for TXCFLG bit of USART_STS register to be set to 1, indicating transmission completion.

Note: TXEN bit cannot be reset during data transmission; otherwise, the data on TX pin will be destroyed, which is because if the baud rate generator stops counting, the data being transmitted will be lost.

20.4.3.2 Single-byte communication

TXBEFLG bit can be cleared by writing USART_DATA register. When the TXBEFLG bit is set by hardware, the shift register will receive the data transferred from the data transmit register, then the data will be transmitted, and the data transmit register will be cleared. The next data can be written in the data register without covering the previous data.

（1） If TXBEIEN in USART_CTRL1 register is set to 1, an interrupt will be generated.

（2） If USART is in the state of transmitting data, write to the data register to save the data to the DATA register, and transfer the data to the shift register at the end of the current data transmission.

（3） If USART is in idle state, write to the data register, put the data into the shift register, start Transmitting data, and set TXBEFLG bit to 1.

（4） When a data transmission is completed and TXBEFLG bit is set, TXCFLG bit will be set to 1; at this time if TXCIEN bit in USART_CTRL1 register is set to 1, an interrupt will be generated.

（5） After the last data is written in the USART_DATA register, before entering the low-power mode or before closing the USART module, wait to set TXCFLG to 1.

20.4.3.3 Break frame

The break frames are considered to receive 0 in a frame period. Setting TXBF bit of USART_CTRL1 register can Transmit a break frame, and the length of the break frame is determined by DBLCFG bit of USART_CTRL1 register. If the TXBF bit is set, after completion of transmission of current data, the TX line will Transmit a break frame, and after completion of transmission of break frame, the TXBF bit will be reset. At the end of the break frame, the transmitter inserts one or two stop bits to respond to the start bit.

Note: If the TXBF bit is reset before transmission of the break frame starts, the break frame will not be transmitted. To transmit two consecutive break frames, the TXBF bit should be set after the stop bit of the previous disconnection symbol.

20.4.3.4 Idle frame

The idle frame is regarded as a complete data frame composed entirely of 1, followed by the start bit of the next frame containing the data. Set TXEN bit of USART_CTRL1 register to 1 and one idle frame can be set before the first data frame.

20.4.4 Receiver

20.4.4.1 Character receiving

During receiving period of USART, RX pin will first introduce the least significant bit of the data. In this mode, USART_ DATA register has a buffer between the internal bus and the receiver shift register. The data is transmitted to the buffer bit by bit. When fully receiving the data, the corresponding receiver register is not empty, then the user can read USART_ DATA.

Receiving configuration steps

- Set UEN bit of USART_CTRL1 register to enable USART.
- Decide the word length by setting DBLCFG bit of USART_CTRL1 register.
- Decide the number of stop bits by setting STOPCFG bit of USART_CTRL2 register.
- If multi-buffer communication is selected, DMA should be enabled in USART_CTRL3 register.
- Set the baud rate of communication in USART_BR register.
- Set RXEN bit of USART_CTRL1 to enable receiving.

Note:

(1) RXEN bit cannot be reset during data receiving period; otherwise, the bytes being received will be lost.

(2) In the process when the receiver is receiving a data frame, if overrun error, noise error or frame error is detected, the error flag will be set to 1.

(3) When data is transferred from the shift register to USART_DATA register, the RXBNEFLG bit of USART_STS will be set by hardware.

(4) An interrupt will be generated if RXBNEIEN bit is set.

(5) In single buffer mode, the RXBNEFLG bit can be cleared by reading USART_DATA register by software or by writing 0.

(6) In multi-buffer mode, after each byte is received, RXBNEFLG bit of USART_STS register will be set to 1, and DMA will read the data register to clear it.

20.4.4.2 Break frame

When the receiver receives a break frame, USART will handle it as receiving a frame error.

20.4.4.3 Idle frame

When the receiver receives an idle frame, USART will handle it as receiving an ordinary data frame; if IDLEIEN bit of USART_CTRL1 is set, an interrupt will be generated.

20.4.4.4 Overrun error

When RXBNEFLG bit of USART_STS register is set to 1 and a new character is received at the same time, an overrun error will be caused. Only after RXEN is reset, can the data be transferred from the shift register to DATA register. RXBNEFLG bit will be set to 1 after receiving the byte. This bit needs to be reset before receiving the next data or serving the previous DMA request; otherwise, an overrun error will be caused.

When an overrun error occurs

- USART_STS OVREFLG bit set to 1.
- The data in DATA register will not be lost.
- The data in the shift register received before will be overwritten, but the data received later will not be saved.
- If RXBNEIEN bit of USART_CTRL1 is set to 1, an interrupt will be generated.
- When OVREFLG bit is set to 1, it means that the data has been lost. There are two possibilities:
  - When RXBNEFLG=1, the previous valid data is still on DATA register, and can be read.
  - When RXBNEFLG=0, there is no valid data in DATA register.
- The OVREFLG bit can be reset through read operation for USART_STS and USART_DATA registers.

20.4.4.5 Noise error

When noise is detected in receiving process of the receiver:

- Set NE flag on the rising edge of RXBNEFLG bit of USART_STS register.
- Invalid data is transmitted from the shift register to USART_DATA register.

20.4.4.6 Frame error

If the stop bit is not received and recognized at the expected receiving time due to excessive noise or lack of synchronization, a frame error will be detected.

When a frame error is detected in receiving process of the receiver:

（1） Set the FEFLG bit of USART_STS register.

（2） Invalid data is transmitted from the shift register to USART_DATA register.

（3） In single byte communication, there is no interrupt, but in multi-buffer communication, an interrupt will be generated by setting the ERRIEN bit of USART_CTRL3 register.

20.4.5 Baud Rate Generator

The baud rate division factor (USARTDIV) is a 16-digit number consisting of 12-digit integer part and 4-digit decimal part. Its relationship with the system clock:

Baud rate=PCLK/16×(USARTDIV)

The system clock of USART2/3 is PCLK1, and that of USART1 is PCLK2. USART can be enabled only after the clock control unit enables the system clock.

20.4.6 Multi-processor Communication

In multi-processor communication, multiple USARTs are connected to form a network. In this network, two devices communicate with each other, and the mute mode can be enabled for other devices not participating in the communication in order to reduce the burden of USART. In mute mode, no receive state bit will be set and all receive interrupts are disabled.

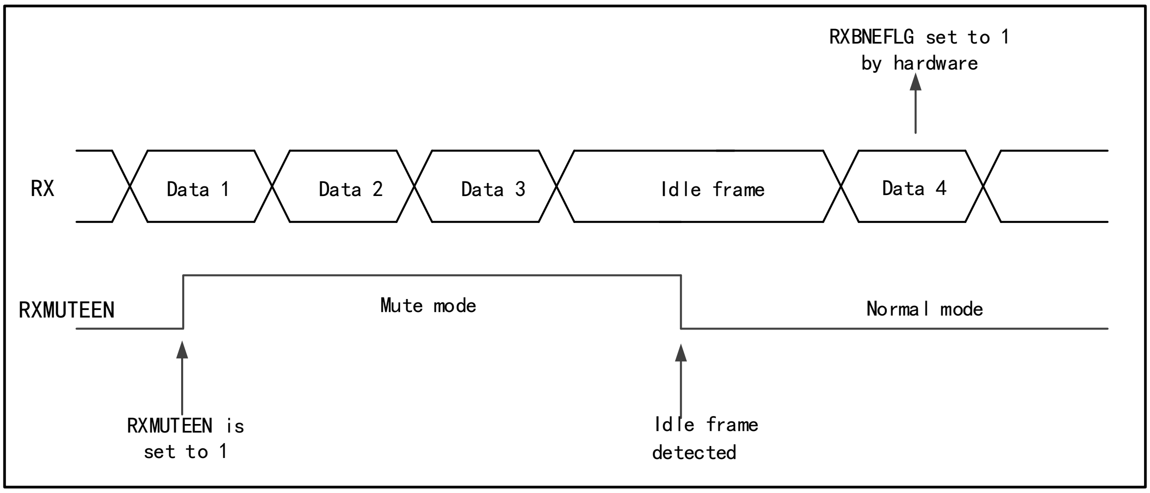
When mute mode is enabled, there are two ways to exit the mute mode:

- WUPMCFG bit is cleared and the bus is idle to exit the mute mode.
- WUPMCFG bit is set and after receiving the address flag, it can exit the mute mode.

Idle bus detection (WUPMCFG=0)

When RXMUTEEN is set to 1, USART enters the mute mode, and it can be waken up from the mute mode when an idle frame is detected, meanwhile, the RXMUTEEN bit will be cleared by the hardware. RXMUTEEN can also be cleared by software.

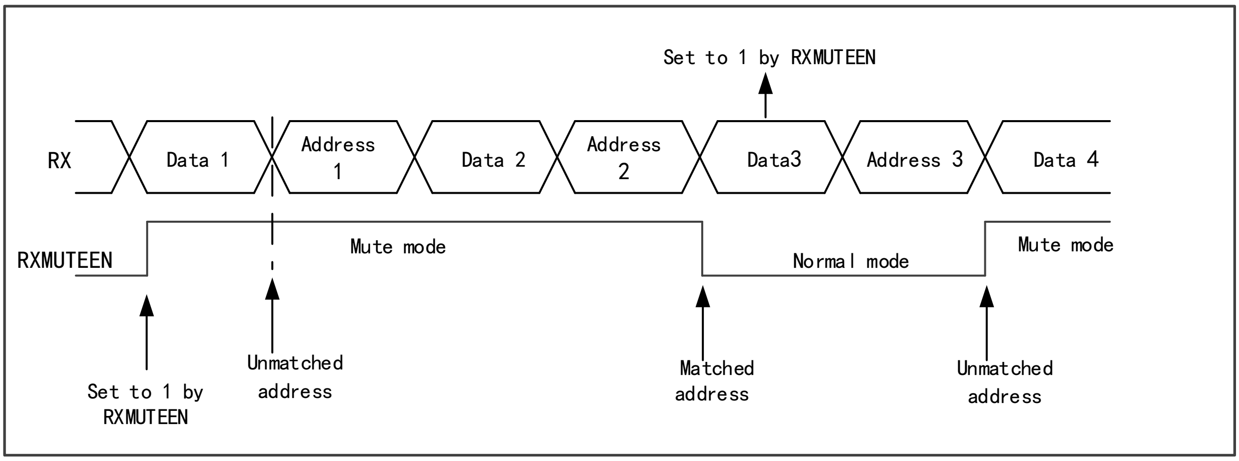
Figure 67 Idle Bus Exit Mute Mode



Address flag detection (WUPMCFG=1)

If the address flag bit is 1, this byte is regarded as the address. The storage address of lower four bits of the address bytes will first be compared with its own address when the receiver receives the address byte. If the addresses do not match, the receiver will enter the mute mode. If the addresses match, the receiver will wake up from the mute mode and be ready to receive the next byte. If the address byte is received again after exiting the mute mode, but the address does not match its own address, the receiver will enter the mute mode again.

Figure 68 Address Flag Exit Mute Mode



20.4.7 Synchronous Mode

The synchronous mode supports full duplex synchronous serial communication in master mode, and has one more signal line USART_CK which can output synchronous clock than the asynchronous mode.

CLKEN bit of USART_CTRL2 register decides whether to enter the synchronous mode.

When USART enters the synchronous mode:

- The LINMEN bit of USART_CTRL2 register, and IREN, HDEN and SCEN bits of USART_CTRL3 register must be cleared.
- The start bit and stop bit of data frame have no clock output.
- Whether the last data bit of data frame generates USART_CK clock is decided by the LBCPOEN bit of USART_CTRL2 register.
- The clock polarity of USART_CK is decided by CPOL bit of USART_CTRL2 register.
- The phase of USART_CK is decided by the CPHA bit of USART_CTRL2.
- The external CK clock cannot be activated when the bus is idle or the frame is disconnected.

Figure 69 USART Synchronous Transmission Example

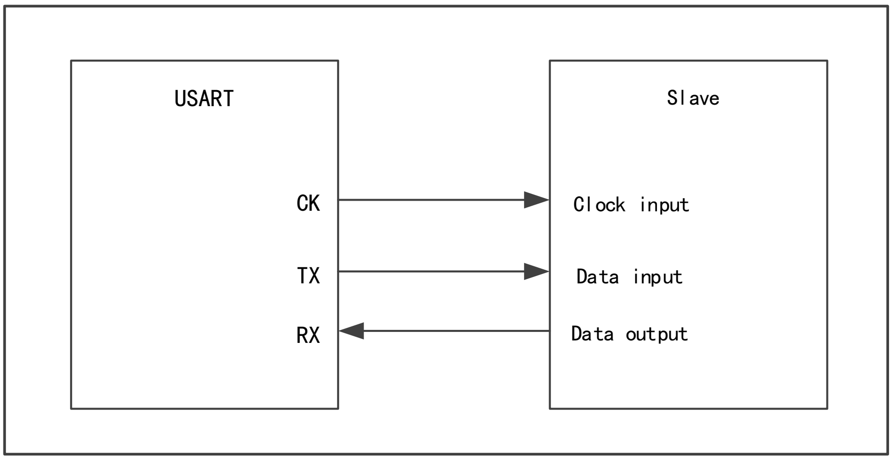


Figure 70 USART Synchronous Transmission Timing Diagram (DBLCFG=0)

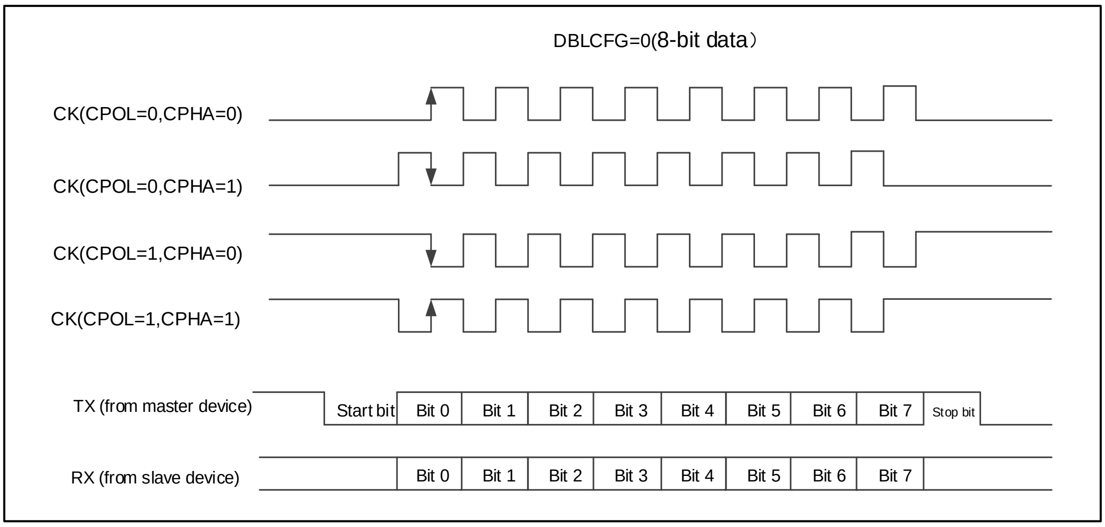
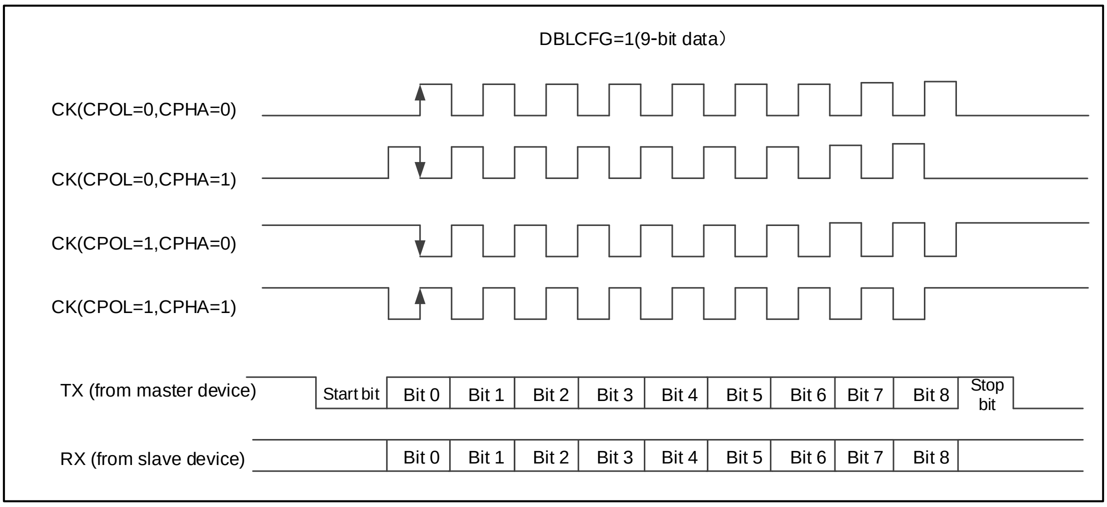


Figure 71 USART Synchronous Transmission Timing Diagram (DBLCFG=1)



20.4.8 LIN Mode

LINMEN bit of USART_CTRL2 register decides whether to enter LIN mode.

When entering LIN mode:

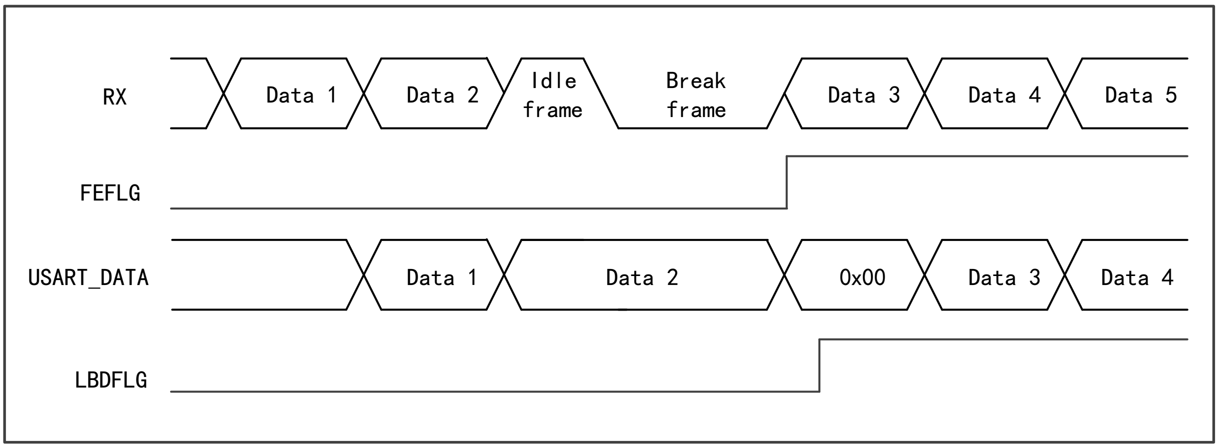
- All data frames are 8 data bits and 1 stop bit.
- The CLKEN bit and STOPCFG bit of USART_CTRL2 register and IREN bit, HDEN bit and SCEN bit of USART_CTRL3 register need to be cleared.

In LIN master mode, USART can generate break frame, and the detection length of break frame can be set to 10 bits and 11 bits through LBDLCFG bit of USART_CTRL2. The break frame detection circuit is independent of USART receiver, and no matter in idle state or in data transmission state, RX pin can detect the break frame, and LBDFLG bit of USART_STS register is set to 1; at this time, if LBDIEN bit of USART_CTRL2 is enabled, an interrupt will be generated.

Detection of break frame in idle state

In idle state, if a break frame is detected on RX pin, the receiver will receive a data frame of 0 and generate FEFLG.

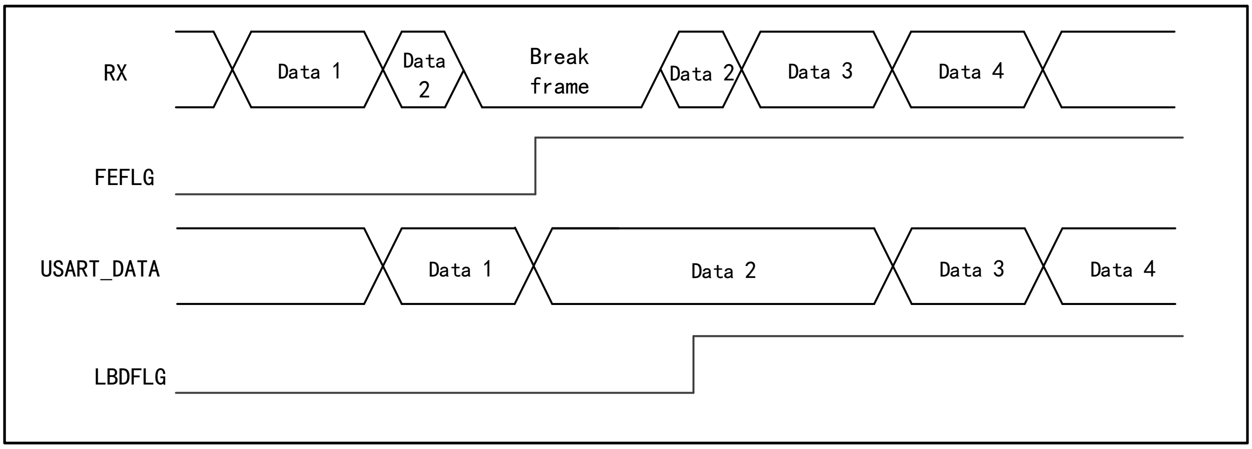
Figure 72 Break Frame Detection in Idle State



Detection of break frame in data transmission state

In the process of data transmission, if the RX pin detects the break frame, the currently transmitted data frame will generate FEFLG.

Figure 73 Break Frame Detection in Data Transmission State



20.4.9 Smart card mode

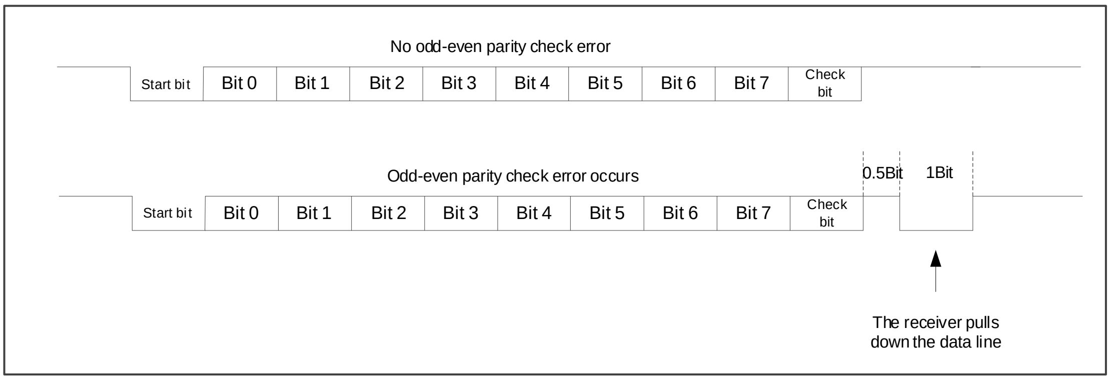
Smart card mode is a single-line half-duplex communication mode. The interface supports ISO7816-3 standard protocol and can control the reading and writing of smart cards that meet the standard protocol.

SCEN bit of USART_CTRL3 register decides whether to enter the smart card mode.

When USART enters the smart card mode:

- The LINMEN bit of USART_CTRL2 register, and IREN and HDEN bits of USART_CTRL3 register must be cleared.
- The data frame format is 8 data bits and 1 check bit, and 0.5 or 1.5 stop bits are used. (To avoid switching between two configurations, it is recommended to use 1.5 stop bits when transmitting and receiving data)
- CLKEN bit of USART_CTRL2 can be set to provide clocks for smart card.
- During the communication, when the receiver detects a parity error, in order to inform the transmitter that the data has not been received successfully, the data line will be pulled down after half a baud rate clock, and keep pulling down for one baud rate clock.
- The break frame has no meaning in smart card mode. A 00h data with frame error will be regarded as a data instead of disconnection symbol.

Figure 74 ISO7816-3 Standard Protocol



20.4.10 Infrared (IrDA SIR) Function Mode

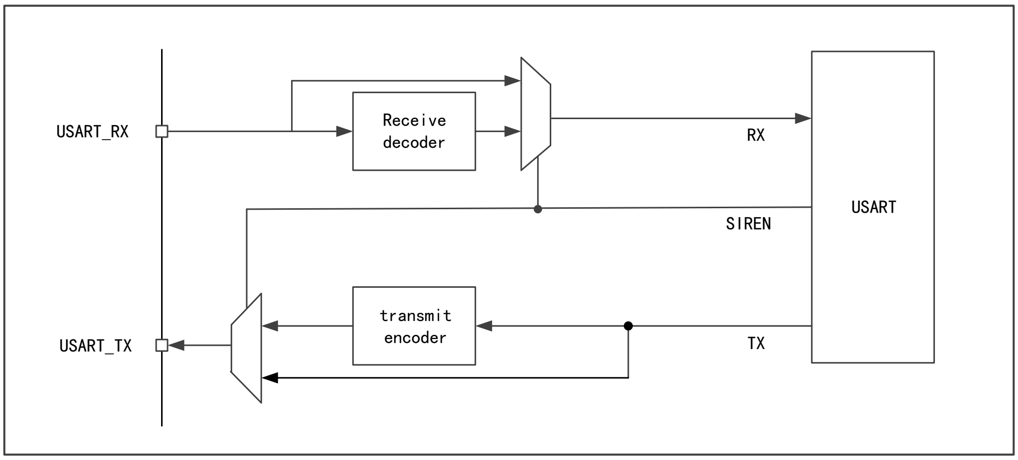
IrDA mode is a half-duplex protocol, transmitting and receiving data can not be carried out at the same time, and the delay between data transmitting and receiving should be more than 10ms.

IREN bit of USART_CTRL3 register decides whether to enter the IrDA mode.

When USART enters the IrDA mode:

- The CLKEN bit, STOPCFG bit and LINMEN bit of USART_CTRL2 register and HDEN bit and SCEN bit of USART_CTRL3 register must be cleared.
- The data frame uses 1 stop bit and the baud rate is less than 115200Hz.
- Using infrared pulse (RZI) indicates logic 0, so in normal mode, its pulse width is 3/16 baud rate cycles. When IrDA is in low power mode, it is recommended that the pulse width be greater than three DIV frequency division clocks so as to ensure that this pulse can be detected by IrDA normally.

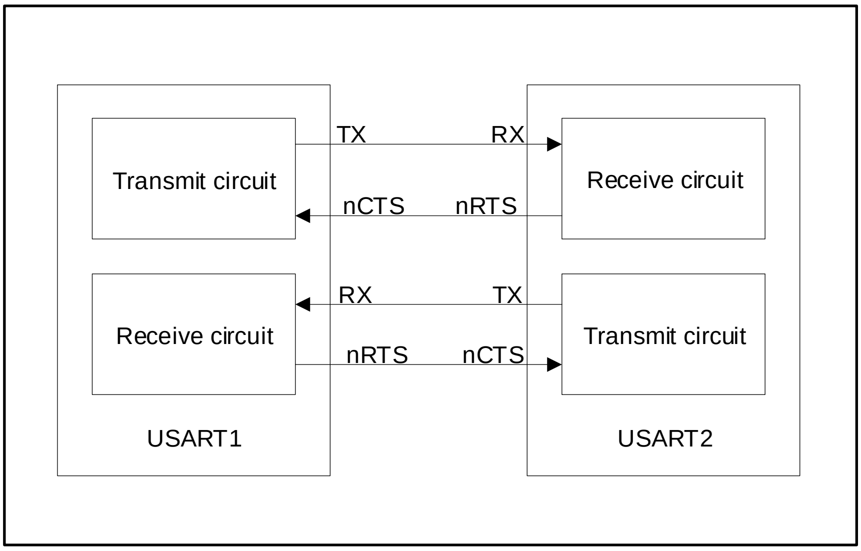
Figure 75 IrDA Mode Block Diagram



20.4.11 Hardware Flow Control

The function of hardware flow control is to control the serial data flow between two devices through nCTS pin and nRTS pin.

Figure 76 Hardware Flow Control between Two USARTs



CTS flow control

CTSEN bit of USART_CTRL3 register determines whether CTS flow control is enabled. If CTS flow control is enabled, the transmitter will detect whether the data frame of nCTS pin can be transmitted. If TXBEFLG bit=0 for USART_STS register and nCTS is pulled to low level, the data frame can be transmitted. If nCTS becomes high during transmission, the transmitter will stop transmitting after the current data frame is transmitted.

RTS flow control

RTSEN bit of USART_CTRL3 register determines whether RTS flow control is enabled. If RTS flow control is enabled, when the receiver receives data, nRTS will be pulled to low level. When a data frame is received, nRTS will becomes high to inform the transmitter to stop transmitting data frame.

20.4.12 DMA Multi-processor Communication

USART can access the data buffer in DMA mode in order to reduce the burden of processors.

Transmission in DMA mode

DMATXEN bit of USART_CTRL3 register determines whether to transmit in DMA mode. When transmitting by DMA, the data in the designated SRAM will be transmitted to the buffer by DMA.

Configuration steps of transmission by DMA:

- Clear the TXCFLG bit of USART_STS register.

- Set the address of SRAM memory storing data as DMA source address.
- Set the address of USART_DATA register as DMA destination address.
- Set the number of data bytes to be transmitted.
- Set channel priority.
- Set interrupt enable.
- Enable DMA channel.
- Wait for TXCFLG bit of USART_STS register to be set to 1, indicating transmission completion.

Receive in DMA mode

DMARXEN bit of USART_CTRL3 register determines whether to receive by DMA. When receiving by DMA, every time one byte is received, the data in the receive buffer will be transmitted to the designated SRAM area by DMA.

Configuration steps of receiving by DMA:

- Set the address of USART_DATA register as DMA source address.
- Set the address of SRAM memory storing data as DMA destination address.
- Set the number of data bytes to be transmitted.
- Set channel priority.
- Set interrupt enable.
- Enable DMA channel.

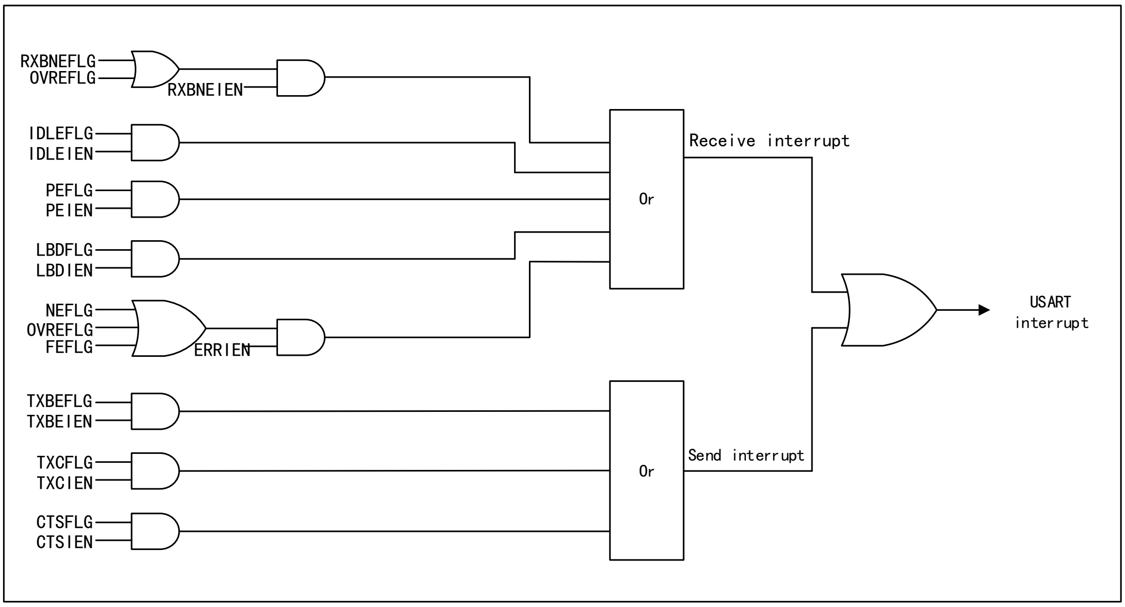
20.4.13 Interrupt Request

Table 84 USART Interrupt Request

Interrupt event		Event flag bit	Enable bit
The receive register cannot be empty		RXBNEFLG	RXBNEIEN
Overload error		OVREFLG	
Line idle is detected		IDLEFLG	IDLEIEN
Odd-even parity error		PEFLG	PEIEN
LIN break frame flag		LBDFLG	LBDIEN
Receiving error in DMA mode	Noise error	NEFLG	ERRIEN
	Overrun error	OVREFLG	
	Frame error	FEFLG	
Data transmit register is empty		TXBEFLG	TXBEIEN
Transmission is completed		TXCFLG	TXCIEN
CTS flag		CTSFLG	CTSIEN

All interrupt requests of USART are connected to the same interrupt controller, and the interrupt requests have logical or relational before they are transmitted to the interrupt controller.

Figure 77 USART Interrupt Mapping



20.4.14 Comparison of USART Supporting Functions

Table 85 Comparison of USART Supporting Functions

USART mode	USART1	USART2	USART3	UART4	UART5
Asynchronous mode	√	√	√	√	√
Hardware flow control	√	√	√	—	—
Multi-buffer communication (DMA)	√	√	√	√	—
Multi-processor communication	√	√	√	√	√
Synchronous	√	√	√	—	—
Smart card	√	√	√	—	—
Half duplex (single-line mode)	√	√	√	√	√
IrDA	√	√	√	√	√
LIN	√	√	√	√	√

Note: "√" means this function is supported, while "—" means that this function is not supported.

20.5 Register Address Mapping

Table 86 USART Register Address Mapping

Register name	Description	Offset address
USART_STS	State register	0x00

Register name	Description	Offset address
USART_DATA	Data register	0x04
USART_BR	Baud rate register	0x08
USART_CTRL1	Control register 1	0x0C
USART_CTRL2	Control register 2	0x10
USART_CTRL3	Control register 3	0x14
USART_GTPSC	Protection time and prescaler register	0x18

20.6 Register Functional Description

20.6.1 State register (USART_STS)

Offset address: 0x00

Reset value: 0x00C0

Field	Name	R/W	Description
0	PEFLG	R	Parity Error Occur Flag 0: No error 1: Parity error occurs In the receiving mode, when a parity error occurs, set to 1 by hardware; This bit can be cleared by software; after setting of RXBNEFLG, first read USART_STS register, and then read USART_DATA register to complete clearing.
1	FEFLG	R	Frame Error Occur Flag 0: No frame error 1: A frame error or disconnection symbol appeared When there is synchronous dislocation, too much noise or disconnection symbol, set to 1 by hardware; This bit can be cleared by software; first read USART_STS register, and then read USART_DATA register to complete clearing.
2	NEFLG	R	Noise Error Occur Flag 0: No noise 1: There is noise error When there is noise error, set to 1 by hardware; This bit can be cleared by software; first read USART_STS register, and then read USART_DATA register to complete clearing.
3	OVREFLG	R	Overrun Error Occur Flag 0: Overrun error 1: Overrun error occurred When the RXBNEFLG bit is set and the data in the shift register is to be transmitted to the receiver register, set to 1 by hardware; This bit can be cleared by software; first read USART_STS register, and then read USART_DATA register to complete clearing.

Field	Name	R/W	Description
4	IDLEFLG	R	IDLE Line Detected Flag 0: Idle bus is not detected 1: Idle bus is detected When idle bus is detected, set to 1 by hardware; This bit can be cleared by software; first read USART_STS register, and then read USART_DATA register to complete clearing.
5	RXBNEFLG	RC_W0	Receive Data Buffer Not Empty Flag 0: The receive data buffer is empty 1: The receive data buffer is not empty When the data register receives the data transmitted by the receiver shift register, set to 1 by hardware; This bit can be cleared by software; read USART_DATA to clear, or write 0 to this bit to clear it.
6	TXCFLG	RC_W0	Transmit Data Complete Flag 0: Sending data is not completed 1: Sending data is completed After the last frame of data is transmitted and the TXBEFLG is set, set to 1 by hardware; This bit can be cleared by software; first read USART_STS register, and then write USART_DATA register to complete clearing; or this bit can be cleared by writing 0 to it.
7	TXBEFLG	R	Transmit Data Buffer Empty Flag 0: The transmit data buffer is not empty 1: The transmit data buffer is empty When the shift register receives the data transmitted by the transmitter data register, set to 1 by hardware; This bit can be cleared by software; write USART_DATA register to complete clearing.
8	LBDFLG	RC_W0	LIN Break Detected Flag 0: LIN disconnection not detected 1: LIN disconnection detected When LIN disconnection is detected, set to 1 by hardware; This bit can be cleared by software; or cleared by writing 0 to this bit.
9	CTSFLG	RC_W0	CTS Change Flag 0: No change on nCTS state line 1: There is change on nCTS state line If the CTSEN bit is set, when switching to the nCTS input, set to 1 by hardware; This bit can be cleared by software; or cleared by writing 0 to this bit.
31:10	Reserved		

20.6.2 Data register (USART_DATA)

Offset address: 0x04

Reset value: 0xXXXX XXXX, X=undefined bit

Field	Name	R/W	Description
8:0	DATA	R/W	Data Value Transmit or receive the data value; read data when receiving data, and write data to the register when transmitting data. When the parity bit is enabled, for 9 data bits, the 8 bit of DATA is parity bit; for 8 data bits, the 7 bit of DATA is parity bit.
31:9	Reserved		

20.6.3 Baud rate register (USART_BR)

Offset address: 0x08

Reset value: 0x0000Reset value: 0x0000

Field	Name	R/W	Description
3:0	FBR[3:0]	R/W	Fraction of USART Baud Rate Divider factor The decimal part of USART baud rate division factor is determined by these four bits.
15:4	IBR[15:4]	R/W	Integer of USART Baud Rate Divider factor The integral part of USART baud rate division factor is determined by these 12 bits.
31:16	Reserved		

20.6.4 Control register 1 (USART_CTRL1)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
0	TXBF	R/W	Transmit Break Frame 0: Not transmit 1: Will transmit This bit can be set by software and cleared by hardware when the stop bit of the break frame is transmitted.
1	RXMUTEEN	R/W	Receive Mute Mode Enable 0: Normal working mode 1: Mute mode This bit is set or cleared by software, or cleared by hardware when wake-up sequence is detected. USART must receive a data before it is put in the mute mode, so that it can be detected and awakened by idle bus. In the wake-up of address flag detection, if the RXBNEFLG bit is set, the RXMUTEEN bit cannot be modified by software.
2	RXEN	R/W	Receive Enable 0: Disable 1: Enable, and start to detect the start bit on RX pin

Field	Name	R/W	Description
3	TXEN	R/W	Transmit Enable 0: Disable 1: Enable Except in smart card mode, if there is a 0 pulse on this bit at any time of transmitting data, an idle bus will be transmitted after the current data is transmitted. After this bit is set, the data will be transmitted after one-bit time.
4	IDLEIEN	R/W	IDLE Interrupt Enable 0: Disable 1: Generate an interrupt when IDLEFLG is set
5	RXBNEIEN	R/W	Receive Buffer Not Empty Interrupt Enable 0: Disable 1: Generate an interrupt when OVREFLG or RXBNEFLG is set
6	TXCIEN	R/W	Transmit Complete Interrupt Enable 0: Disable 1: Generate an interrupt when TXCFLG is set
7	TXBEIEN	R/W	Transmit Buffer Empty Interrupt Enable 0: Interrupt generation is disabled 1: Generate an interrupt when TXBEFLG is set
8	PEIEN	R/W	Parity Error interrupt Enable 0: Interrupt generation is disabled 1: Generate an interrupt when PEFLG is set
9	PCFG	R/W	Odd/Even Parity Configure 0: Even parity check 1: Odd parity check The selection will not take effect until the current transmission of bytes is completed.
10	PCEN	R/W	Parity Control Enable 0: Disable 1: Enable If this bit is set, a check bit will be inserted in the most significant bit when transmitting data; when receiving data, check whether the check bit of the received data is correct. The check control will not take effect until the current transmission of bytes is completed.
11	WUPMCFG	R/W	Wakeup Method Configure 0: Idle bus wakeup 1: Address tag wakeup
12	DBLCFG	R/W	Data Bits Length Configure 0: 1 start bit, 8 data bits, n stop bits 1: 1 start bit, 9 data bits, n stop bits This bit cannot be modified during transmission of data.
13	UEN	R/W	USART Enable 0: USART frequency divider and output are disabled 1: USART module is enabled
31:14	Reserved		

20.6.5 Control register 2 (USART_CTRL2)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	ADDR[3:0]	R/W	USART Device Node Address Setup This bit is valid only in the mute mode of multiprocessor communication, and decides to enter the mute mode or wake up according to whether the detected address tags are consistent.
4	Reserved		
5	LBDLCFG	R/W	LIN Break Detection Length Configure 0: 10 bits 1: 11 bits
6	LBDIEN	R/W	LIN Break Detection Interrupt Enable 0: Disable 1: Generate an interrupt when LBDFLG bit is set
7	Reserved		
8	LBCPOEN	R/W	Last Bit Clock Pulse Output Enable 0: Not output from CK 1: Output from CK This bit is valid only in synchronous mode; this bit does not exist on UART4 and UART5.
9	CPHA	R/W	Clock Phase Configure This bit indicates on the edge of which clock sampling is conducted 0: The first 1: The second This bit is valid only in synchronous mode; this bit does not exist on UART4 and UART5.
10	CPOL	R/W	Clock Polarity Configure The state of CK pin when USART is in idle state 0: Low level 1: High level This bit is valid only in synchronous mode; this bit does not exist on UART4 and UART5.
11	CLKEN	R/W	Clock Enable (CK pin) 0: Disable 1: Enable This bit does not exist on UART4 and UART5.
13:12	STOPCFG	R/W	STOP Bit Configure 00: 1 stop bit 01: 0.5 stop bit 10: 2 stop bit 11: 1.5 stop bit This bit does not exist on UART4 and UART5.
14	LINMEN	R/W	LIN Mode Enable 0: Disable 1: Enable

Field	Name	R/W	Description
31:15	Reserved		

Note: These three bits (CPOL, CPHA and LBCPOEN) cannot be changed after transmission is enabled.

20.6.6 Control register 3 (USART_CTRL3)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
0	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable; when DMARXEN is set and one among FEFLG, OVREFLG or NEFLG is set, an interrupt will be generated.
1	IREN	R/W	IrDA Function Enable 0: Disable 1: Enable
2	IRLPEN	R/W	IrDA Low-power Mode Enable 0: Normal mode 1: Low-power mode
3	HDEN	R/W	Half-duplex Mode Enable 0: Disable 1: Enable
4	SCNACKEN	R/W	NACK Transmit Enable During Parity Error in Smartcard Function 0: NACK is not transmitted 1: Transmit NACK This bit does not exist on UART4 and UART5.
5	SCEN	R/W	Smartcard Function Enable 0: Disable 1: Enable This bit does not exist on UART4 and UART5.
6	DMARXEN	R/W	DMA Receive Enable 0: Disable 1: Enable This bit does not exist on UART4 and UART5.
7	DMATXEN	R/W	DMA Transmit Enable 0: Disable 1: Enable This bit does not exist on UART4 and UART5.
8	RTSEN	R/W	RTS Hardware Flow Control Function Enable 0: Disable 1: Enable RTS interrupt RTS: Require To Send, which is output signal, indicating it has been ready to receive. Request is made to receive data only when there is space in the receive buffer; when data can be received, RTS output is pulled to low level. This bit does not exist on UART4 and UART5.

Field	Name	R/W	Description
9	CTSEN	R/W	CTS Hardware Flow Control Function Enable 0: Disable 1: Enable CTS: Clear To Send, which is input signal When CTS input signal is at low level, the data can be transmitted; otherwise, the data cannot be transmitted; if CTS signal is pulled to high during data transmission, the data transmission will be stopped after the data transmission is completed; if write operation is performed for the data register when CTS is high, the data will not be transmitted until CTS is valid. This bit does not exist on UART4 and UART5.
10	CTSIEN	R/W	CTS Interrupt Enable 0: Disable 1: Generate an interrupt when CTSFLG is set This bit does not exist on UART4 and UART5.
31:11	Reserved		

20.6.7 Protection time and prescaler register (USART_GTPSC)

Offset address: 0x18

Reset value: 0x0000

Field	Name	R/W	Description
7:0	PSC	R/W	Prescaler Factor Setup Divide the frequency and provide clock for the system clock respectively; in different working modes, the valid bits of PSC have difference, specifically as follows: In infrared low-power mode: PSC[7:0] is valid. 00000000: Reserved 00000001: 1 divided frequency 00000010: 2 divided frequency …… 11111111: 255 divided frequency In infrared normal mode: PSC can only be set to 00000001 In smart card mode: PSC[7:5] invalid, PSC[4:0] valid 00000: Reserved 00001: 2 divided frequency 00010: 4 divided frequency 00011: 6 divided frequency …… 11111: 62 divided frequency This bit does not exist on UART4 and UART5.
15:8	GRDT	R/W	Guard Time Value Setup After transmitting data, TXCFLG can be set after the protection time; the time unit is baud clock; it can be applied to smart card mode; this bit does not exist on UART4 and UART5.
31:16	Reserved		

21 Internal Integrated Circuit Interface (I2C)

21.1 Full Name and Abbreviation Description of Terms

Table 87 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Serial Data	SDA
Serial Clock	SCL
System Management Bus	SMBus
Clock	CLK
Serial Clock High	SCLH
Serial Clock Low	SCLL
Address Resolution Protocol	ARP
Negative Acknowledgement	NACK
Packet Error Checking	PEC
Address Resolution Protocol	ARP

21.2 Introduction

I2C is a short-distance bus communication protocol. In physical implementation, I2C bus is composed of two signal lines (SDA and SCL) and a ground wire. These two signal lines can be used for bidirectional transmission.

- Two signal lines, SCL clock line and SDA data line. SCL provides timing for SDA, and SDA transmits/receives data in series
- Both SCL and SDA signal lines are bidirectional
- The ground is common when the two systems use I2C bus for communication

21.3 Main Characteristics

（1） Multi-master function

（2） The master can generate the clock, start bit and stop bit

（3） Slave function

- Programmable I2C address detection
- Double-address mode
- Detection stop bit

（4） 7-bit and 10-bit addressing mode

（5） Response to broadcast

（6） Two communication speeds

- Standard mode
- Fast mode

（7） Programmable clock extension

（8） State flag

- Transmitter/Receiver mode flag
- Flag for end of byte transmission
- Flag of busy bus

（9） Error flag

- Arbitration loss
- Acknowledgment error
- Wrong start bit or stop bit detected

（10） Interrupt source

- Address/Data communication succeeded
- Error interrupt

（11） Support DMA function

（12） Programmable PEC

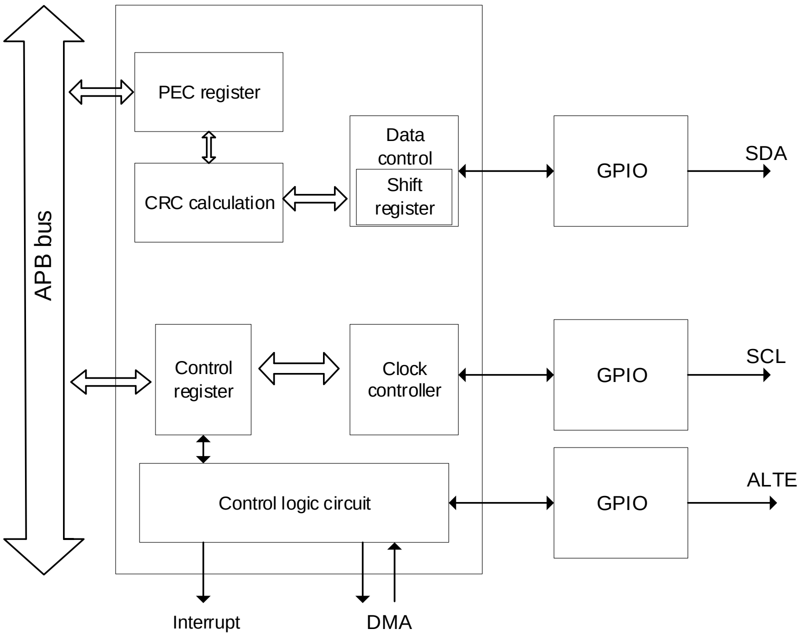
- Final transmission in transmission mode
- PEC error check after the last byte is received

（13） SMBus specific function

- Hardware PEC
- Address resolution protocol

21.4 Structure Block Diagram

Figure 78 I2C Function Structure Diagram



The interface can be configured to the following modes:

- Slave transmitting
- Slave receiving
- Master transmitting
- Master receiving

In the initial state of I2C interface, the working mode is slave mode. After I2C interface sends the start signal, it will automatically switch from slave mode to master mode.

21.5 Functional Description

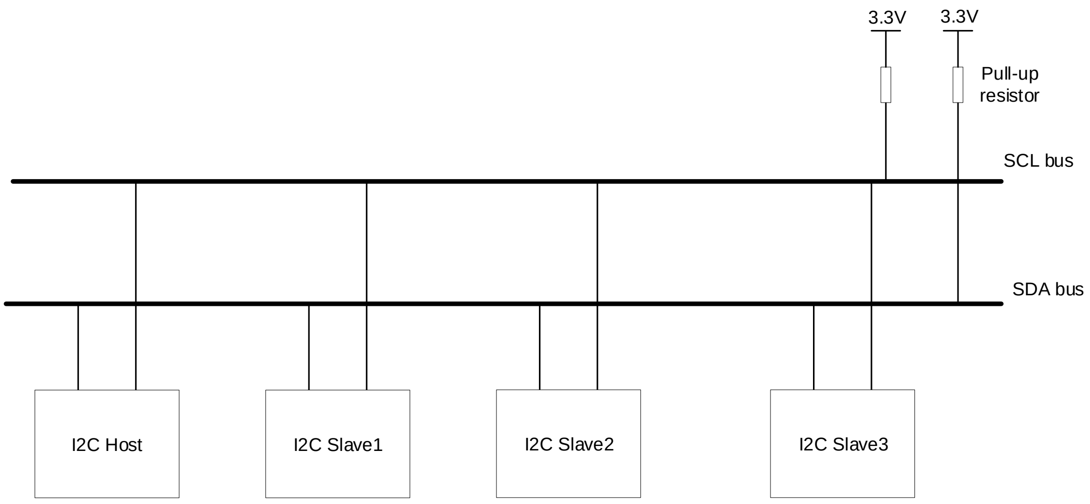
Table 88 Description of Special Terms of I2C Bus

Special terms	Instruction
Transmitter	Device transmitting data to the bus
Receiver	Device receiving data from the bus
Master	Device that initiates data transmission, generates clock signals and ends data transmission
Slave	Device addressed by master
Multiple masters	Multiple masters that control the bus at the same time without destroying information
Synchronous	The process of synchronizing the clock signals between two or more devices
Arbitration	If more than one master tries to control the bus at the same time, only one master can control the bus, and the information of the controlled master will not be destroyed

21.5.1 I2C Physical Layer

The commonly used connection modes between I2C communication devices are shown in the figure below:

Figure 79 Commonly Used I2C Communication Connection Diagram



Characteristics of physical layer:

（1） Bus supporting multiple devices (signal line shared by multiple devices), which, in I2C communication bus, can connect multiple communication masters and communication slaves.

（2） An I2C bus only uses two bus lines, namely, a bidirectional serial data line (SDA) and a serial clock line (SCL). The data line is used for data transmission, and the clock line is used for synchronous receiving and transmission of data.

（3） Each device connected to the bus has an independent address (seven or ten bits), and the master addresses and accesses the slave device according to the address of the device.

（4） The bus needs to connect the pull-up resistor to the power supply. When I2C bus is idle, the output is in high-impedance state. When all devices are idle, the output is in high-impedance state, and the pull-up resistor pulls the bus to high level.

（5） Three communication modes: Standard mode (up to 100KHz), fast mode (up to 400KHz), and fast mode plus (up to 1MHz).

（6） When multiple masters use the bus at the same time, to prevent the data conflict, the bus arbitration mode is adopted to determine which device occupies the bus.

（7） Can program setup and hold time, and program the high-level time and low-level time of SCL in I2C.

21.5.2 I2C Protocol Layer

Characteristics of protocol layer

（1） Data is transmitted in the form of frame, and each frame is composed of 1 byte (8 bits).

（2） In the rising edge phase of SCL, SDA needs to keep stable and SDA changes during the period when SCL is low.

（3） In addition to data frame, I2C bus also has start signal, stop signal and acknowledge signal.

- Start bit: During the stable high level period of SCL, a falling edge of SDA starts transmission.
- Stop bit: During the stable high level period of SCL, a rising edge of SDA stops transmission.
- Acknowledge bit: Used to indicate successful transmission of one byte. After the bus transmitter (regardless of the master or slave) transmits 8-bit data, SDA will release (from output to input). During the ninth clock pulse, the receiver will pull down SDA to respond to the received data.

I2C communication reading and writing process

Figure 80 Master Writes Data to Slave

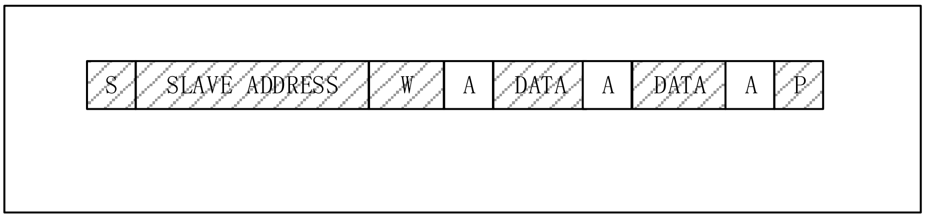
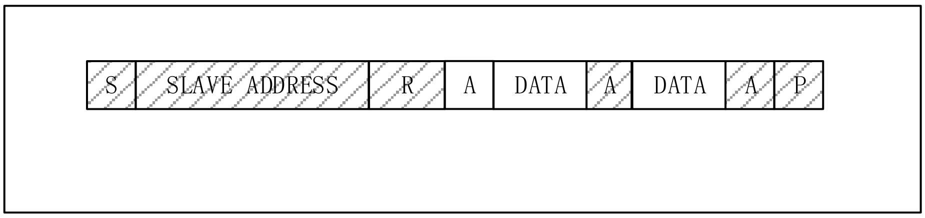


Figure 81 Master Reads Data from Slave



Remarks:

（1） ▨: This data is transferred from master to slave

（2） S: Start signal

（3） SLAVE ADDRESS: Slave address

（4） □: This data is transferred from slave to master

（5） R/W: Selection bit of transmission direction

（6） 1 means read

（7） 0 means write

（8） P: Stop signal

After the start signal is generated, all slaves will wait for the slave address signal transmitted by the master. In I2C bus, the address of each device is unique. When the address signal matches the device address, the slave will be selected, and the unselected slave will ignore the future data signal.

When the master direction is writing data

After broadcasting the address and receiving the acknowledge signal, the master will transmit data to the slave, the data length is one byte, and every time the master transmits one byte of data, it needs to wait for the answer signal transmitted by the slave. After all the bytes have been transmitted, the master will transmit a stop signal (STOP) to the slave, indicating that the transmission is completed.

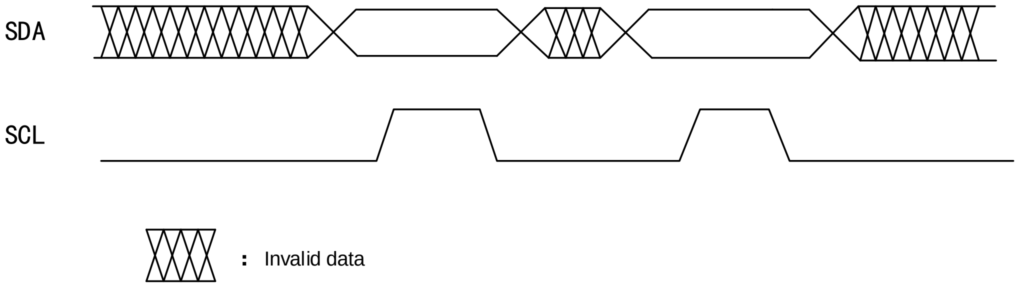
When the master direction is reading data

After broadcasting the address and receiving the acknowledge signal, the slave will transmit the data to the master. The size of the data package is 8 bits. Every time the slave sends one byte of data, it needs to wait for the acknowledge signal of the master. When the master wants to stop receiving data, it needs to return a non-acknowledge signal to the slave, then the slave will stop transmitting the data automatically.

21.5.3 Data Validity

In the process of data transmission, the data on SDA line must be stable when the clock signal SCL is at high level. Only when the SCL is at the low level, can the level state of SDA be changed, and the bit transmission of each data needs a clock pulse.

Figure 82 SDA Timing Diagram



21.5.4 Start and Stop Signals

All data transfer must have start signal (START) and stop signal (STOP).

Figure 83 START signal is defined as: when SCL is at high level, SDA will convert from high level to low level

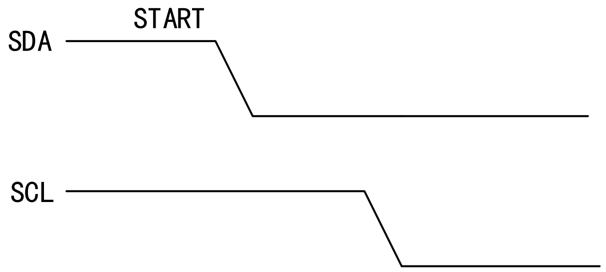
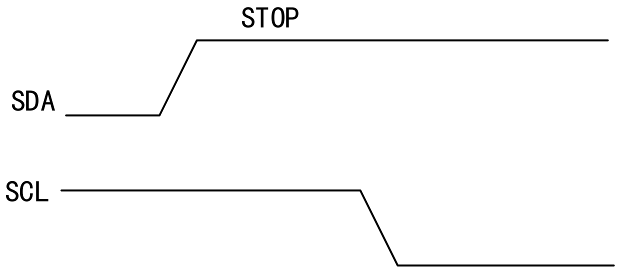


Figure 84 STOP signal is defined as: when SCL is at high level, SDA will convert from low level to high level



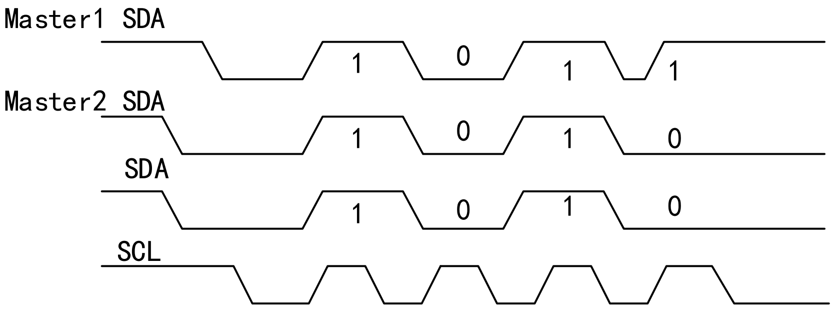
21.5.5 Arbitration

Arbitration is also used to solve the bus control conflict in case of multiple masters. The arbitration process takes place on the master and has nothing to do with the slave.

The master can start transmission only when the bus is idle. Two masters may generate an effective START signal on the bus within the shortest hold time of the START signal. In this situation, it is required by arbitration to decide which master completes the transmission.

Arbitration is conducted by bit. During each arbitration, when SCL is high, each master will check whether the SDA level is the same as that transmitted by itself. The arbitration process needs to last for many bits. Theoretically, if two masters transmit exactly the same content, they can successfully transmit without arbitration failure. If one master transmits high level, but it is detected that SDA is at low level, an arbitration failure error will occur, the SDA output of the master will be closed, and the other master will complete its own transmission.

Figure 85 SDA Timing Diagram



Note: Master 1 arbitration failure

21.5.6 SMBus Specific Function

System management bus (SMBus) is a simple single-end double-wire bus, which can meet the requirements of lightweight communication.

SMBus is commonly used in computer motherboard, mainly for power transmission ON/OFF instructions. SMBus is the derivative bus of I2C. It is mainly used for communication of low-bandwidth devices on computer motherboard, and power-related chip.

Address resolution protocol

SMBus specification includes an address resolution protocol, which can realize dynamic address assignment. Dynamic recognition hardware and software enable the bus to support hot plugging, and the bus devices will be automatically identified and assigned with a unique address.

SMBus alarm

SMBus alarm is an optional signal with an interrupt line for pins that are sacrificed to extend their control ability.

21.5.7 Error Flag Bit

Table 89 The following several error flag bits exist in I2C communication

Error flag bit	Description of error flag bit
Answer error flag bit (AEFLG)	No answer received
Bus error flag bit (BERRFLG)	An external stop or start condition is detected
Arbitration loss flag bit (ALFLG)	Arbitration loss is detected by the interface
Overrun/Underrun error flag bit (OVRURFLG)	In slave mode, the received data is not read out, the next data has arrived, and an overrun error occurs. The transmitting data clock has arrived, but the data has not been written into the DATA register, and an underrun error occurs.
Timeout or Tow error flag bit (TTEFLG)	SCL is pulled down for more than a certain time
PEC comparison error flag bit (PECEFLG)	CRC values are not equal

21.5.8 Message Error Check (PEC)

I2C module has a PEC module, which checks the message of I2C data by CRC-8 calculator. The CRC-8 polynomial used by the calculator is: C(x)= X^8+X^2+X+1.

When PECEN bit is set to 1 and PEC function is enabled, PEC module will calculate all data transmitted by I2C bus, including address data.

21.5.9 DMA Mode

According to the software process of I2C, when the transmitter register is empty or the receiver register is full, MCU needs to write or read bytes, then we can complete the operation more quickly through the DMA function of I2C.

DMA transmission

Set the DMAEN bit in I2C_CTRL2 register to enabe the DMA mode. When the transmitter register is empty (TXBEFLG is set to 1), the data will be directly loaded from the memory area to the DATA register through DMA.

DMA receiving

Set DMAEN in I2C_CTRL2 register to enable DMA mode. When the receiving register is full (RXBNEFLG is set to 1), DMA will transmit DATA register data to the set storage area.

21.5.10 I2C Interrupt

Table 90 I2C Interrupt Request

Interrupt event	Event flag bit	Interrupt control bit
Transmitting start bit completed	STARTFLG	EVIEN
Transmission completed/Address matching address signal	ADDRFLG	
10-bit address head segment transmission completed	ADDR10FLG	
Received stop signal	STOPFLG	
Data byte transmission completed	BTCFLG	
Receive buffer not empty	RXBNEFLG	EVIEN and BUFIEN
Transmit buffer empty	TXBEFLG	
Bus error	BERRFLG	ERRIEN
Arbitration loss	ALFLG	
Answer failed	AEFLG	
Overrun/Underrun	OVRURFLG	
PEC error	PECEFLG	
Timeout or Tlow error	TTEFLG	
SMBus alert	ALERTEN	

21.6 Register Address Mapping

Table 91 I2C Register Address Mapping

Register name	Description	Offset address
I2C_CTRL1	Control register 1	0x00
I2C_CTRL2	Control register 2	0x04
I2C_SADDR1	Slave address register 1	0x08
I2C_SADDR2	Slave address register 2	0x0C
I2C_DATA	Data register	0x10
I2C_STS1	State register 1	0x14
I2C_STS2	State register 2	0x18
I2C_CLKCTRL	Master clock control register	0x1C
I2C_RISETMAX	Maximum rising time register	0x20
I2C_SWITCH	I2C Switch register	0x100

21.7 Register Functional Description

21.7.1 Control register 1 (I2C_CTRL1)

Offset address: 0x00

Reset value: 0x0000

Field	Name	R/W	Description
0	I2CEN	R/W	I2C Enable 0: Disable 1: Enable
1	SMBEN	R/W	SMBus Mode Enable 0: I2C mode 1: SMBus mode
2	Reserved		
3	SMBTCFG	R/W	SMBus Type Configure 0: SMBus device 1: SMBus master
4	ARPEN	R/W	ARP Enable 0: Disable 1: Enable If SMBTCFG=0, use the default address of SMBus device If SMBTCFG=1, use the primary address of SMBus
5	PECEN	R/W	PEC Enable 0: Disable 1: Enable
6	SRBEN	R/W	Slave Responds Broadcast Enable 0: Disable 1: Enable Note: The broadcast address is 0x00
7	CLKSTRETCHD	R/W	Slave Mode Clock Stretching Disable 0: Enable 1: Disable In slave mode, enabling extending the low-level time of the clock can avoid overrun and underrun errors.
8	START	R/W	Start Bit Transfer This bit can be set to 1 and cleared by software; when transmitting the start bit or I2CEN=0, it is cleared by hardware. 0: Not transmit 1: transmit
9	STOP	R/W	Stop Bit Transfer This bit can be set to 1 or cleared by software; when sending the stop bit, it is cleared by hardware; when timeout error is detected, it is set to 1 by hardware. 0: Not send 1: Send
10	ACKEN	R/W	Acknowledge Transfer Enable

Field	Name	R/W	Description
			This bit can be set to 1 or cleared by software; when I2CEN=0, it is cleared by hardware. 0: Not send 1: Send
11	ACKPOS	R/W	Acknowledge /PEC Position Configure This bit can be set to 1 or cleared by software; when I2CEN=0, it is cleared by hardware. 0: When receiving current byte, whether sending NACK/ACK, whether PEC is in shift register 1: When receiving next byte, whether sending NACK/ACK and whether PEC is in the next byte of shift register
12	PEC	R/W	Packet Error Check Transfer Enable This bit can be set to 1 or cleared by software; when sending PEC, or sending the start bit and stop bit, or when I2CEN=0, it is cleared by hardware. 0: Disable 1: Enable
13	ALERTEN	R/W	SMBus Alert Enable This bit can be set to 1 or cleared by software; when I2CEN=0, it is cleared by software. 0: Release the SMBAlert pin to make it higher, and remind to send the response address header immediately after sending the NACK signal 1: Drive SMBAlert pin to make it lower, and remind to send the response address header immediately after sending the ACKEN signal
14	Reserved		
15	SWRST	R/W	Software Configure I2C under Reset State 0: Not reset 1: Reset; before I2C reset, ensure that I2C pin is released and the bus is in idle state.

21.7.2 Control register 2 (I2C_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
5:0	CLKFCFG	R/W	I2C Clock Frequency Configure The clock frequency is the clock of I2C module, namely, the clock input from APB bus. 0: Disable 1: Disable 2: 2MHz ... 50: 50MHz Greater than 100100: Disable. Minimum clock frequency of I2C bus: the standard mode is 1MHz, and the fast mode is 4MHz.

Field	Name	R/W	Description
7:6	Reserved		
8	ERRIEN	R/W	Error Interrupt Enable 0: Disable 1: When the position 1 of any of the following state register is enabled, the interrupt will be generated: SMBALTFLG, TTEFLG, PECEFLG, OVRURFLG, AEFLG, ALFLG, and STS1_BERRFLG
9	EVIEN	R/W	Event Interrupt Enable 0: Disable 1: When the position 1 of any of the following state registers is enabled, the interrupt will be generated: STARTFLG, ADDRFLG, ADDR10FLG, STOPFLG, BTCFLG, TXBEFLG is set to 1 and BUFIEN is set to 1, RXBNEFLG is set to 1 and BUFIEN is set to 1.
10	BUFIEN	R/W	Buffer Interrupt Enable 0: Disable 1: Enable; when the bit of any of the following state register is set to 1, the interrupt will be generated: TXBEFLG and RXBNEFLG
11	DMAEN	R/W	DMA Requests Enable 0: Disable 1: When TXBEFLG=1 or RXBNEFLG=1, enable DMA request
12	LTCFG	R/W	DMA Last Transfer Configure Configure whether the EOT of the next DMA is the last transmission received, and only used for the master receiving mode. 0: No 1: Yes
15:13	Reserved		

21.7.3 Slave mode address register 1 (I2C_SADDR1)

Offset address: 0x08

Reset value: 0x0000

Field	Name	R/W	Description
0	ADDR[0]	R/W	Slave Address Setup When the address mode is 7 bits, the bit is invalid; when the address mode is 10 bits, this bit is the 0 bit of the address.
7:1	ADDR[7:1]	R/W	Slave Address Setup Slave address 7:1 bit
9:8	ADDR[9:8]	R/W	Slave Address Setup When the address mode is 7 bits, the bit is invalid; when the address mode is 10 bits, this bit is the 9:8 bit of the address.
14:10	Reserved		
15	ADDRLEN	R/W	Slave Address Length Configure 0: 7-bit address mode 1: 10-bit address mode

21.7.4 Slave address register 2 (I2C_SADDR2)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
0	ADDRNUM	R/W	Slave Address Number Configure In the slave 7-bit address mode, it can be configured to identify the single-address mode and double-address mode; only ADDR1 is identified in single-address mode; both ADDR1 and ADDR2 can be identified in double-address mode Single or double address registers can be identified in 7-bit address mode, specifically as follows: 0: Identify one address (ADDR1) 1: Identify two addresses (ADDR1 and ADDR2)
7:1	ADDR2[7:1]	R/W	Slave Dual Address Mode Address Setup bit7:1 of the address in double-address mode
15:8	Reserved		

21.7.5 Data register (I2C_DATA)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
7:0	DATA	R/W	Data Register In I2C transmission mode, write the data to be transmitted to this register; in I2C receiving mode, read the received data from this register.
15:8	Reserved		

21.7.6 State register 1 (I2C_STS1)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
0	STARTFLG	R	Start Bit Sent Finished Flag 0: Not transmit 1: Transmitted When the start bit is transmitted, this bit can be set to 1 by hardware; this bit can be cleared after the software first reads STS1 register and then writes the DATA register; when I2CEN=0, it can be cleared by hardware.
1	ADDRFLG	R	Address Transfer Complete /Receive Match Flag Whether the matching address is received in slave mode: 0: Not received 1: Received Whether finishing sending the address in master mode: 0: Not completed 1: Completed The bit is set to 1 by hardware; this bit can be cleared after the software first reads STS1 register and then reads STS2 register; when I2CEN=0, it can be cleared by hardware.
2	BTCFLG	R	Byte Transfer Complete Flag 0: Not completed 1: Completed

Field	Name	R/W	Description
			When receiving data, if failing to read the data received in DATA register, and a new data is received then, set to 1 by hardwre; When sending data, if the DATA register is empty, to send the data in the shift register, set to 1 by hardware. This bit can be cleared after the software first reads STS1 register, and then reads or writes the DATA register; this bit can be cleared by hardware by sending a start bit or stop bit during the transmission, or when I2CEN=0.
3	ADDR10FLG	R	10-Bit Address Header Sent Flag 0: Not transmit 1: Transmitted The bit is set to 1 by hardware; this bit can be cleared after the software first reads STS1 register and then writes the DATA register; when I2CEN=0, it can be cleared by hardware.
4	STOPFLG	R	Stop Bit Detection Flag 0: Not detected 1: Detected If ACKEN=1, after one answer, when the slave detects the stop bit on the bus, it will be set to 1 by hardwre; This bit can be cleared after the software first reads STS1 register and then writes CTRL1 register; when I2CEN=0, it can be cleared by hardware.
5	Reserved		
6	RXBNEFLG	R	Receive Buffer Not Empty Flag 0: The receive buffer is empty 1: The receive buffer is not empty This bit can be set to 1 by hardware when there is data in DATA register; When BTCFLG is set to 1, since the data register is still full, the RXBNEFLG bit cannot be cleared by reading DATA register; This bit can be cleared after the software reads and writes DATA register; when I2CEN=0, it can be cleared by hardware.
7	TXBEFLG	R	Transmit Buffer Empty Flag 0: The transmit buffer is not empty 1: The transmit buffer is empty This bit can be set to 1 by hardware when the content of DATA register is empty; When the software writes the first data to the DATA register, it will immediately move the data to the shift register, then the data in the DATA register is empty and this bit cannot be cleared; This bit can be cleared after the software writes data to DATA register; after sending the start bit or stop bit, or when I2CEN=0, it can be cleared by hardware.
8	BERRFLG	RC_W0	Bus Error Flag 0: No bus error 1: Bus error occurred Bus error means exception of start bit or stop bit; when an error is detected, this bit can be set to 1 by hardware; this bit can be

Field	Name	R/W	Description
			cleared after the software writes 0; when I2CEN=0, it can be cleared by hardware.
9	ALFLG	RC_W0	Master Mode Arbitration Lost Flag 0: No arbitration loss 1: In case of arbitration loss, I2C interface will automatically switch back to slave mode "Arbitration loss in master mode" means the master loses the control of buses; this bit is set to 1 by hardware; this bit can be cleared after the software writes 0; when I2CEN=0, it is cleared by hardware.
10	AEFLG	RC_W0	Acknowledge Error Flag 0: No acknowledgment error 1: Acknowledgment error occurred This bit can be set to 1 by hardware; this bit can be cleared after the software writes 0; when I2CEN=0, it can be cleared by hardware.
11	OVRURFLG	RC_W0	Overrun/Underrun Flag 0: Not occur 1: Occurred This bit can be set to 1 by hardware when CLKSTRETCHD=1 and one of the following conditions is met: (1) In the slave receiving mode, when the data in the DATA register is not read out, but a new data is received (this data will be lost), overrun occurs; (2) In the slave transmission mode, no data is written in the data register but it still needs to send data (the same data is transmitted twice), and then underrun occurs. This bit can be cleared by writing 0 by software; or be cleared by hardware when I2CEN=0.
12	PECEFLG	RC_W0	PEC Error in Reception Flag 0: No PEC error: when ACKEN=1, after receiving PEC, the receiver will return ACKEN 1: There is PEC error; regardless of the value of ACKEN, as long as PEC is received, the receiver will return NACK This bit can be cleared by writing 0 by software; or be cleared by hardware when I2CEN=0.
13	Reserved		
14	TTEFLG	RC_W0	Timeout or Tlow error flag (Timeout or Tlow Error Flag) 0: No timeout error 1: When a timeout error occurs, in slave mode, the slave is reset and the bus is released; in master mode, the hardware sends the stop bit. This bit can be set to 1 by hardware when timeout error occurs in any of the following situations: (1) SCL maintains low level for more than 25ms; (2) SCL low-level extension time of the main device is more than 10ms; (3) SCL low-level extension time of the slave device is more than 25ms.

Field	Name	R/W	Description
			This bit can be cleared by writing 0 by software; or be cleared by hardware when I2CEN=0.
15	SMBALTFLG	RC_W0	SMBus Alert Occur Flag 0: SMBus master mode, without alarm; SMBus slave mode, without alarm, SMBAlert pin level unchanged 1: SMBus master mode, with an alarm generated on the pin; SMBus slave mode, receiving an alarm, causing SMBAlert pin level to become low This bit can be set to 1 by hardware; this bit can be cleared after the software writes 0; when I2CEN=0, it can be cleared by hardware.

21.7.7 State register 2 (I2C_STS2)

Offset address: 0x18

Reset value: 0x0000

Field	Name	R/W	Description
0	MSFLG	R	Master Slave Mode Flag 0: Slave mode 1: Master mode This bit can be set to 1 by hardware when I2C is configured as master mode; This bit can be cleared by hardware when one of the following conditions is met: (1) Stop bit is generated (2) Bus arbitration is lost (3)I2CEN=0
1	BUSBSYFLG	R	Bus Busy Flag 0: The bus is idle (no communication) 1: The bus is busy (in the progress of communication) This bit can be set to 1 by hardware when SDA or SCL is at low level; cleared by hardware after the stop bit is generated.
2	TRFLG	R	Transmitter / Receiver Mode Flag 0: The device is in receiver mode (read) 1: The device is in transmitter mode (write) Decide the bit value according to R/W bit; This bit can be cleared by hardware when one of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) Bus arbitration loss (4)I2CEN=0
3	Reserved		
4	GENCALLFLG	R	Slave Mode Received General Call Address Flag 0: Failed to receive the broadcast address 1: Received broadcast address

Field	Name	R/W	Description
			This bit can be set to 1 by hardware; and be cleared by hardware when one of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3)I2CEN=0
5	SMBDADDRFLG	R	SMBus Device Received Default Address Flag in Slave Mode 0: Failed to receive the default address 1: Received the default address when ARPEN=1 This bit can be set to 1 by hardware; and be cleared by hardware when one of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) I2CEN=0
6	SMMHADDR	R	SMBus Device Received Master Header Flag in Slave Mode 0: Failed to receive the master head address 1: Received the master head address when SMBTSEL=1 and ARPEN=1 This bit can be set to 1 by hardware; and be cleared by hardware when one of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) I2CEN=0
7	DUALADDRFLG	R	Slave Mode Received Dual Address Match Flag 0: The received address matches the content of ADDR1 register 1: The received address matches the content of ADDR2 register This bit can be set to 1 by hardware; and be cleared by hardware when one of the following conditions is met: (1) Stop bit is generated (2) Repeated start bit is generated (3) I2CEN=0
15:8	PECVALUE	R	Save Packet Error Checking Value When PECEN=1, the internal PEC value is saved in PECVALUE.

21.7.8 Master clock control register (I2C_CLKCTRL)

Offset address: 0x1C

Reset value: 0x0000

Field	Name	R/W	Description
11:0	CLKS [11:0]	R/W	Clock Setup in Fast/Standard Master Mode In I2C standard mode or SMBus mode: T_{high}=CLKS × T_{PCLK1} T_{low}=CLKS × T_{PCLK1} In I2C fast mode: When FDUTYCFG=0: T_{high}=CLKS×T_{PCLK1} T_{low}=2×CLKS× T_{PCLK1} When FDUTYCFG=1: T_{high}=9 × CLKS × T_{PCLK1}

Field	Name	R/W	Description
			T_{low}=16 × CLKS × T_{PCLK1}
13:12	Reserved		
14	FDUTYCFG	R/W	Fast Mode Duty Cycle Configure Here define the duty cycle=t_{low}/t_{high} 0: SCLK duty cycle is 2 1: SCLK duty cycle is 16/9
15	SPEEDCFG	R/W	Master Mode Speed Configure 0: Standard mode 1: Fast mode

21.7.9 Maximum rising time register (I2C_RISETMAX)

Offset address: 0x20

Reset value: 0x0002

Field	Name	R/W	Description
5:0	RISETMAX	R/W	Master Mode Maximum Rise Time in Fast/Standard Mode The time unit is T_{PCLK1}, and RISETMAX is the maximum rising time of SCL plus 1.
15:6	Reserved		

21.7.10 I2C switch register (I2C_SWITCH)

Offset address: 0x100

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	SWITCH	R/W	I2C Switch 0: I2C uses I2C1/I2C2 1: I2C uses I2C3
31:1	Reserved		

22 Internal Integrated Circuit Interface (I2C3)

22.1 Introduction

I2C bus is a two-wire serial interface, which consists of serial data line (SDA) and serial clock (SCL). These lines transmit information between devices connected to the bus. Each device is identified by a unique address and can work as a "transmitter" or "receiver", depending on the function of the device. When performing data transmission, the device can also be regarded as a master or slave. The master is the device that initiates data transmission on the bus and generates clock signals to enable the transmission. At this time, any device that is addressed is considered as slave.

I2C can run in standard mode (data rate is from 0 to 100 kb/s), fast mode (data rate is less than or equal to 400 kb/s), fast mode plus (data rate is less than or equal to 1000 kb/s), and high-speed mode (data rate is less than or equal to 3.4 kb/s).

I2C can only communicate with the devices in these modes as long as they are connected to the bus. Besides, the high-speed mode and fast-mode devices are downward compatible. For example, in a hybrid speed bus system, the high-speed mode devices can communicate with the devices in fast mode and standard mode. In the I2C bus system of 0-100kb/s, the fast-speed mode device can communicate with the standard mode device.

An example of high-speed mode devices is LCD display, high-bit count ADC and high-capacity EEPROM. These devices usually need to transmit mass data. Most maintenance and control applications, namely, the common use of I2C bus, usually operate at 100 kHz (in standard and fast-speed mode).

22.2 Functional Description

- Double-line I2C serial interface - consists of a serial data line (SDA) and a serial clock (SCL).
- Three speeds:
  - Standard mode (0 to 100 Kb/s)
  - Fast-speed mode (≤ 400 Kb/s) or fast mode +(≤ 1000 Kb/s)
  - High-speed mode (≤ 3.4 Mb/s)
- Clock synchronization
- Master or slave I2C operation
- 7 or 10-bit address
- 7 or 10-bit combined transmission
- Bulk transfer mode
- Ignore CBUS address (used for sharing I2C old address of I2C bus)
- Sending and receiving buffers

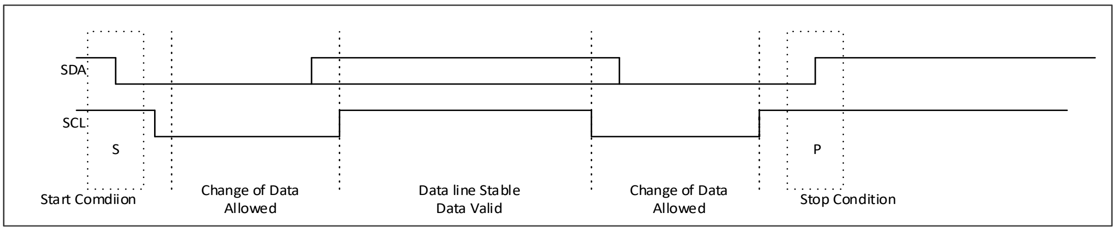
- Bit and byte waiting at the speeds of processing all buses
- Programmable SDA hold time (tHD; DAT)
- Bus clear characteristics
- Dedicated transmit FIFO and receive FIFO; FIFO depth is 8, and FIFO bit width is 8 bits

22.2.1 I2C Protocol

Start (START) and stop (STOP) conditions

When the bus is idle, both SCL and SDA signals are pulled up through the external pull-up resistor on the bus. If the master wants to start transmission on the bus, it shall issue the start condition. This is defined as the high-low conversion of SDA signal, and SCL is 1. If the master wants to terminate the transmission, it shall issue the stop condition. This is defined as the low-high conversion of SDA line, and SCL is 1. The figure below shows the timing of start and stop conditions. When data is transmitted on the bus, SDA line must be stable if SCL is 1.

Figure 86 Start (START) and Stop (STOP) Conditions



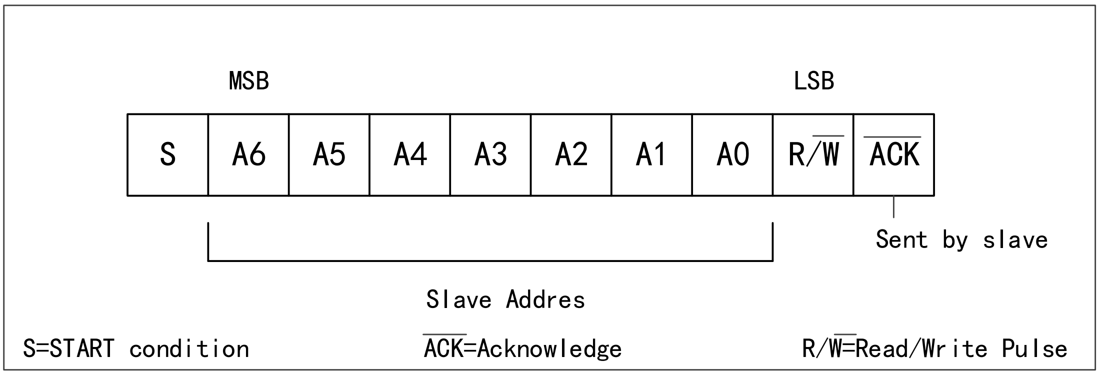
Addressing slave protocol

Two address formats: 7-bit address format and 10-bit address format

7-bit address format

In the 7-bit address format, when bit 0 (R/W) is set to 0, the master writes to the slave. Whe the bit 0 (R/W) is set to 1, the master reads from the slave.

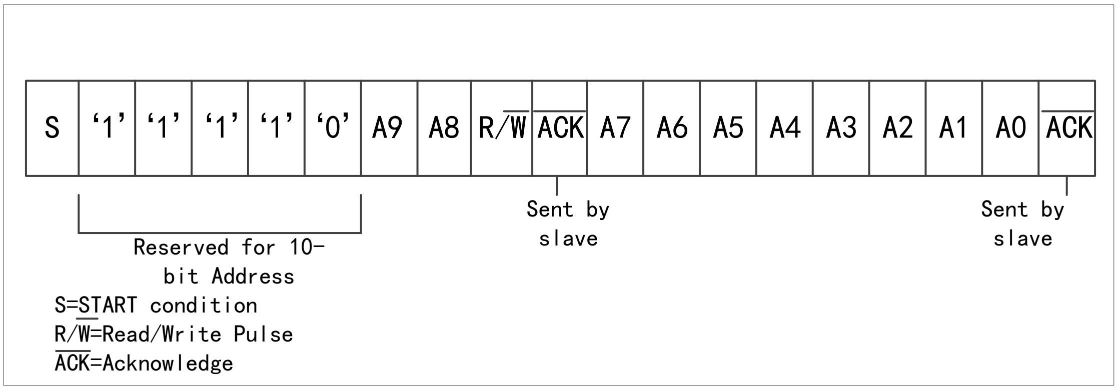
Figure 87 7-bit Address Format



10-bit address format

In the 10 bit address format, two bytes are needed to transmit the 10 bit address. The transmission of the first byte contains the following bit definitions. The first five bits (bits 7:3) notify the slave that this is a 10-bit transmission, they are followed by the next two bits (bits 2:1) setting the slave address bits 9:8, and LSB bit (bit 0) as R/W bit. The second byte of the transmission sets the bits 7:0 of the slave address.

Figure 88 10-bit Address Format



Bit definition of I2C in the first byte

Table 92 I2C Bit Definition

Slave address	R/W bit	Description
0000 000	0	Broadcast call address. I2C puts the data in the receive buffer and triggers a broadcast call interrupt
0000 000	1	Start bit
0000 001	X	CBUS address. I2C ignores these addresses
0000 010	X	Reserved
0000 011	X	Reserved
0000 1xx	X	High-speed master (master) code
1111 1xx	X	Reserved
1111 0xx	X	10-bit slave address.
0001 000	X	Reserved
0001 100	X	Reserved
1100 001	X	Reserved

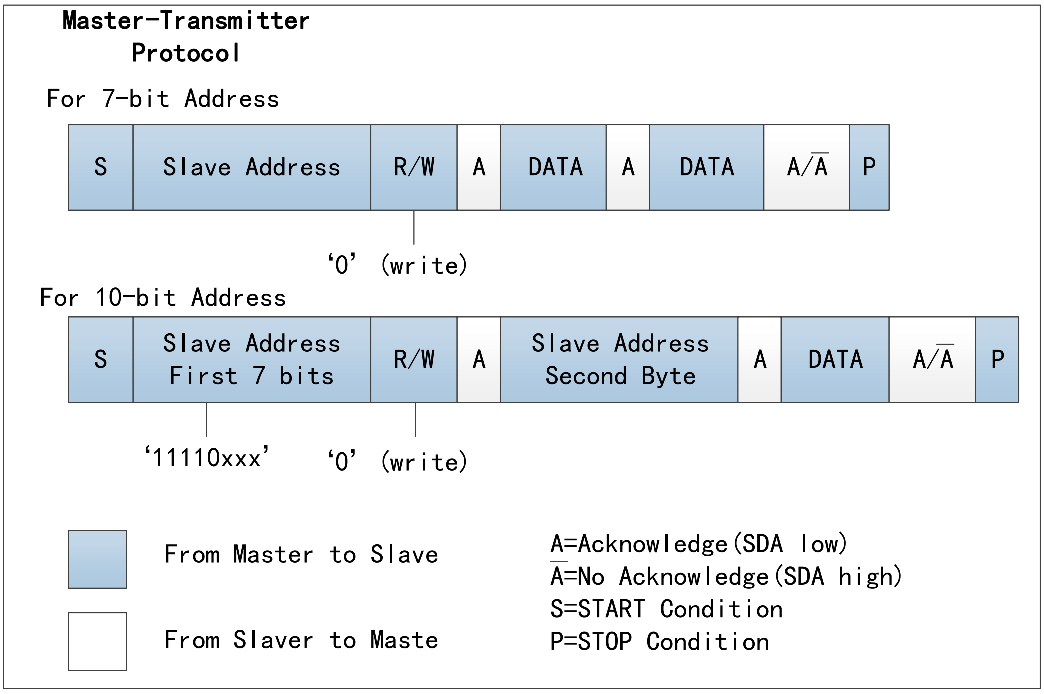
Transmit and receive protocol

As the master transmitter or master receiver, the master can initiate the data transmission and receiving between buses. The slave responds to the requests from the master to send data to the bus or receive data from the bus.

Master transmitter and slave receiver

All data are transmitted in byte format, and the number of bytes of each data for transmission is unrestricted. After the master sends address and R/W bit or the master sends data byte to the slave, the slave receiver must respond with acknowledgment signal (ACK). When the slave receiver does not respond to the ACK pulse, the master will can abort the transfer by sending a stop condition. The slave must keep the SDA line in high bit so that the master can abort the transfer. See the following figure for details.

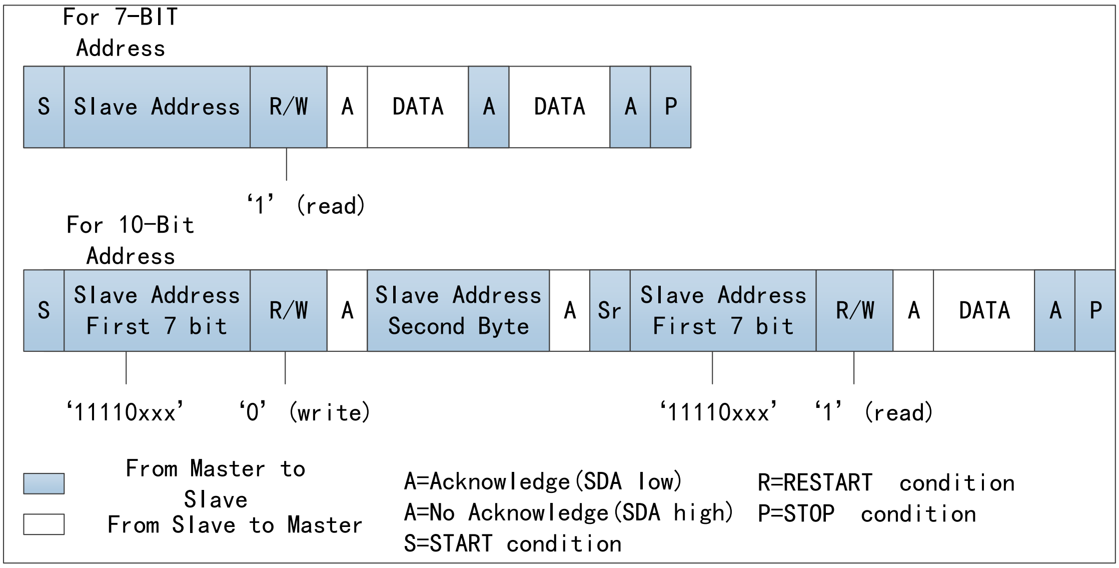
Figure 89 Master Transmitter and Slave Receiver



Master receiver and slave transmitter

If the master is receiving the data as shown in the figure below, after receiving one byte of data (except the last byte), the master will use an acknowledgment impulse to respond to the slave transmitter. Not sending acknowledgment pulse is a way for the master receiver to inform the slave transmitter that this is the last byte. The slave transmitter abandons the SDA line after detecting no acknowledgment (NACK) so that the master transmitter can issue the stop (STOP) condition.

Figure 90 Master Receiver Protocol



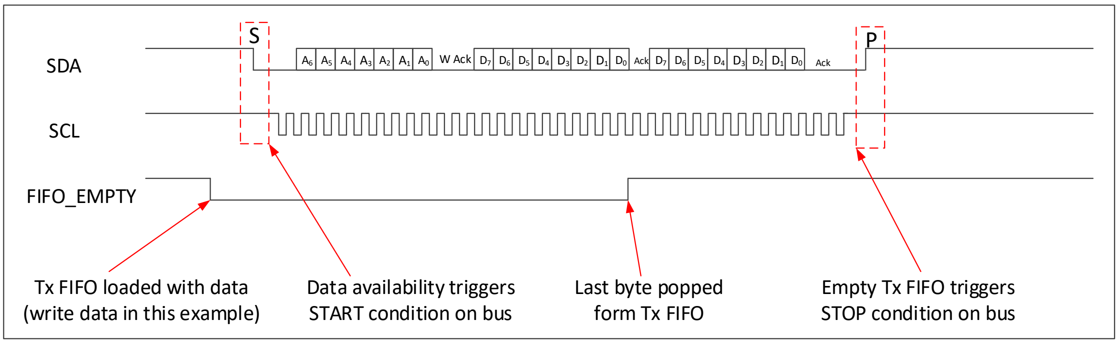
The slave transmitter abandons the SDA line after detecting no acknowledgment (NACK) so that the master transmitter can issue the stop (STOP) condition. This is the same as the start condition, except that it occurs after the ACK pulse. When running in the master mode, I2C can communicate with the same slave through transmission in different directions.

TXFIFO management and start (START), stop (STOP), and restart (RSTAEN)

Whenever TXFIFO becomes empty, the component will generate a stop (STOP) point on the bus. If the restart function is enabled, the component will generate a restart when the transmission direction in the TXFIFO command changes from read to write, and vice versa. If restart is not enabled, a stop will be generated in this case, and then a start will be generated.

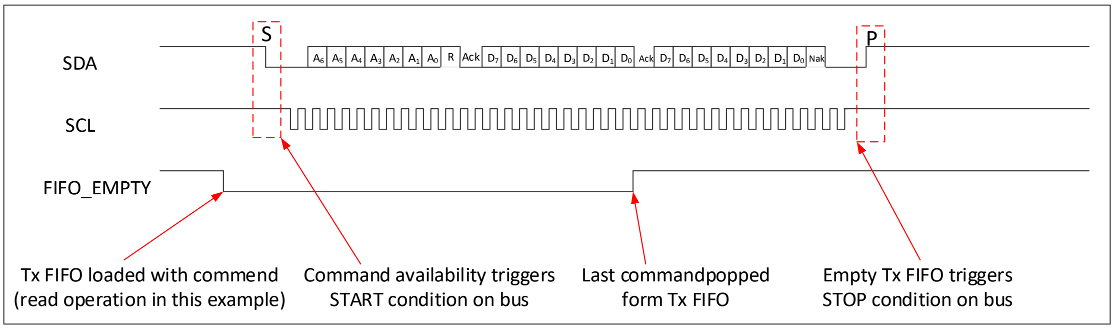
The following timing diagram describes the behavior of I2C when TXFIFO becomes empty when it is running as the master transmitter.

Figure 91 I2C Timing Diagram "Taking TXFIFO as Master Transmitter"


S
P
SDA
SCL
FIFO_EMPTY
A6 A5 A4 A3 A2 A1 A0 W Ack D7 D6 D5 D4 D3 D2 D1 D0 Ack D7 D6 D5 D4 D3 D2 D1 D0 Ack
Tx FIFO loaded with data (write data in this example)
Data availability triggers START condition on bus
Last byte popped form Tx FIFO
Empty Tx FIFO triggers STOP condition on bus

The following timing diagram describes the behavior of I2C when TXFIFO becomes empty when it is running as the master receiver.

Figure 92 I2C Timing Diagram "Taking TXFIFO as Master Receiver"


SDA
SCL
FIFO_EMPTY
S
P
A6 A5 A4 A3 A2 A1 A0 R Ack D7 D6 D5 D4 D3 D2 D1 D0 Ack D7 D6 D5 D4 D3 D2 D1 D0 Nak
Tx FIFO loaded with commend (read operation in this example)
Command availability triggers START condition on bus
Last commandpopped form Tx FIFO
Empty Tx FIFO triggers STOP condition on bus

22.2.2 Multi-master Arbitration

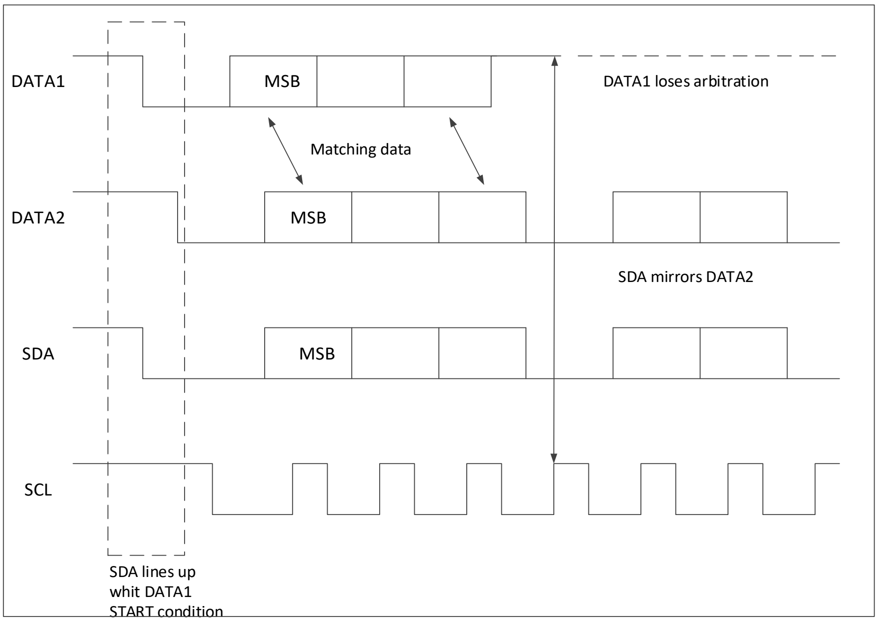
I2C bus protocol allows multiple masters to reside on the same bus. When there are two masters on the same I2C bus, if both masters attempt to control the bus by generating start condition at the same time, there will be an arbitration program.

Once a master controller (such as MCU) controls the bus, no other master controller can control the bus until the first master controller sends a stop condition and the bus is idle.

Arbitration takes place on SDA line, while SCL line is 1. When another master transmits 0, the master transmits 1, the arbitration is lost and its data output phase is closed. The master that lost the arbitration can continue to generate the clock until the end of the byte transmission. If both masters are processing the same slave, arbitration may enter the data phase.

Once it is detected that it has lost arbitration with another master, I2C will stop generating SCL (IC_CLK_oe).

Figure 93 Timing of Arbitration of Two Master Controllers on the Bus



For high-speed mode, since each master is programmed with a unique high-speed main code, arbitration cannot enter the data phase. The 8-bit code is defined by the system designer and is set by writing it into the high-speed master mode code address register HSMC. Since these codes are unique, only one master code can win arbitration, which occurs at the end of high-speed master code transmission. The control of the bus is determined by the address or master code and the data transmitted by the competing master, so there is neither a central master nor any priority order on the bus.

Arbitration is not allowed between restart condition and data bit, stop condition and data bit, restart condition and stop condition.

22.2.3 Clock Synchronization

When two or more masters attempt to transmit information on the bus at the same time, they must arbitrate and synchronize the SCL clock. Data is only valid in the high cycle of the SCL clock. Clock synchronization is performed by connecting lines to SCL signal. When the master converts the SCL clock to 0, the master starts to count the low time of the SCL clock, and converts the SCL clock signal to 1 at the beginning of the next clock cycle. However, if another master maintains the SCL line at 0, the master will enter a high wait state until the SCL clock row is converted to 1.

Then, all masters will report their high time, and the master with the shortest high time converts the SCL line to 0. Then, the master calculates its low time, and the one with the longest low time forces the other master to enter a high wait state. Therefore, a synchronized SCL clock will be generated, as shown in Figure xxx. Or the slave can maintain the SCL line in a low position to slow down the timing on the I2C bus.

22.2.4 IC_CLK Frequency Configuration

When I2C is configured as standard speed (SS), fast speed (FS)/fast mode plus (FM+) or high-speed (HS) master control, the registers must be set before any I2C bus service occurs to ensure correct I/O timing.

Registers in different modes are shown in the table below:

Table 93 Registers in Different Modes

Mode	Register
Standard speed (SS) mode	SSCHC SSCLC
Fast speed (FS)/fast speed mode plus (FM+)	HSCHC HSCLC
High speed (HS)	HSCHC HSCLC

Minimum high and low count in SS, FS, FM+ and HS mode

When I2C is working as I2C master, in sending and receiving transmission:

（1） The value of SSCLC and FSCLC registers must be greater than LSSSL+7.

（2） The value of SSCHC and FSCHC registers must be greater than LSSSL+5.

（3） If this component is programmed to support HS, the value of HSCLC register must be greater than HSSSL+7.

（4） If this component is programmed to support HS, the value of HSCLC register must be greater than HSSSL+5.

Details about I2C high and low counts are as follows:

（1） The minimum value of IC_*_SPKLEN+7 of *_LCNT register is the time required to drive SDA after I2C is on the negative edge of SCL.

（2） The minimum value of IC_*_SPKLEN+5 of *_HCNT register is the time required to collect SDA samples because I2C is in the high-cycle period of SCL.

（3） I2C adds a loop to the programmed *_LCNT value to generate the low cycle of SCL clock, which is because of the count logic of SCL low count to (*_ LCNT+1).

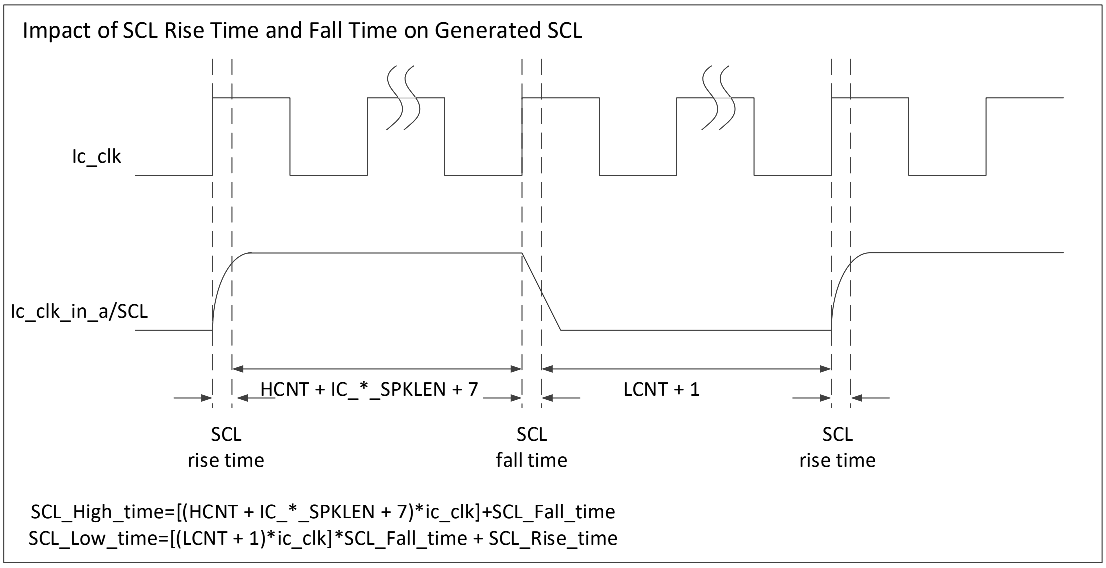
（4） I2C adds IC_*_SPKLEN+7 loop to the programmed *_HCNT value to generate the high cycle of SCL clock, which is because of the following factors.

- Count ogic of SCL high count to (*_HCNT+1).
- The digital filtering applied to SCL line will lead to delay of spklen+2IC_clk cycle,
- The filtering includes metastable removal and programmable peak suppression of SDA and SCL edges.
- When I2C drives SCL from 1 to 0, the SCL high time is completed, and internal logic delay of three IC_CLK cycles will be generated. Therefore, the minimum SCL low time that I2C can achieve is nine IC_CLK cycles (7+1+1), while the minimum SCL high time is thirteen IC_CLK cycles (6+1+3+3).

Note: The total high and low time of SCL generated by I2C master is also affected by the rising time and falling time of SCL line, as shown in the figure and equation. It should be noted that the rising and falling time parameters of SCL vary with external factors, such as:

- IO driver characteristics
- Pull-up resistance
- Total capacitance on SCL line, etc.

Figure 94 Rising and Falling Time Parameter Configuration Diagram


Impact of SCL Rise Time and Fall Time on Generated SCL
Ic_clk
Ic_clk_in_a/SCL
HCNT + IC_*_SPKLEN + 7
LCNT + 1
SCL
rise time
SCL
fall time
SCL
rise time
SCL_High_time=[(HCNT + IC_*_SPKLEN + 7)*ic_clk]+SCL_Fall_time
SCL_Low_time=[(LCNT + 1)*ic_clk]*SCL_Fall_time + SCL_Rise_time

22.2.5 Operation Mode

22.2.5.1 Slave mode operation

Initial Configuration

To use I2C as a slave device, perform the following steps:

（1） Disable I2C by writing "0" to bit 0 of CTRL1 register

（2） Write SLAADDR register (bit 9:0) to set slave address. This is the address of the I2C response

（3） Write to CTRL0 register to specify which type of addressing (7 or 10 bits, set by bit 3) is supported by writing "0" to bit 6 (IC_SLAVE_DISABLE) and writing "0" to bit 0 (MST), and enable I2C in slave only mode.

（4） I2C is enabled by writing "1" to bit 0 of CTRL1 register.

Single-byte slave transmitter operation

When another I2C master on the bus addresses I2C and requests data, I2C will act as a slave transmitter, and the following steps will occur:

（1） Another I2C master starts I2C transmission, and its address matches the slave address in SLAADDR register of I2C.

（2） I2C confirms the sending address and identifies the transmission direction to indicate that it is a slave transmitter.

（3） I2C enables the RRIF interrupt (No. 5 bit of the RIS register) and keeps the SCL line low. It is in wait state until the software responds. If the RRIF interrupt has been masked because the INTEN [5] register (RRIE bit field) is set to 0, it is recommended that the hardware and/or software timing program be used to indicate that the CPU reads the RIS register periodically.

- The read operation indicating that RIS [5] (RRIF bit field) is set to 1 must be regarded as the equivalent operation of the enabled RRIF interrupt.
- Then the software must operate to satisfy I2C transmission.
- The timing interval used should be 10 times of the fastest SCL clock cycle that I2C can handle. For example, for 400 kb/s, the timing interval is 25μs.

（4） If there is any data in the transmit FIFO before the read request is received, I2C enables the Tx ABRT interrupt (Bit 6 of the RIS register) to refresh the old data from the transmit FIFO. If the Tx-ABRT interrupt is masked, since the INTEN[6] register (TAIE bit field) is set to 0, it is recommended that the timing program (described in the previous step) or similar program be used again to read the RIS register.

- Read indicating that Bit 6 (TAIF) is set to 1 must be regarded equivalent to the enabled TAIF interrupt.
- The software does not need further operation.
- The timing interval used should be similar to the time interval described for RIS[5] register in the previous step.

（5） The software writes the data to the DATA register (by writing '0' to Bit 8).

（6） Before the program, the software must clear the RRIF and TAIF interrupts (Bit 5 and Bit 6 respectively) of RIS register. If the RRIF and/or TAIF interrupts are masked, when the RRIF or TAIF bit is read to 1, clear of the RIS register will have been performed.

（7） I2C releases SCL and transmits bytes.

（8） The master can retain the I2C mainline by issuing the restart (RSTAEN) condition or release the mainline by sending a stop (STOP) condition.

Single-byte slave receiver operation

When another I2C master on the mainline addresses I2C and sends data, I2C acts as a slave receiver, to perform the following steps:

（1） Another I2C master starts an I2C transmission with the address in SLAADDR register that matches I2C slave address.

（2） I2C confirms the address transmitted and identifies the direction of the transmission to indicate that I2C acts as a slave receiver.

（3） I2C receives the bytes transmitted and puts them in the receive buffer.

（4） I2C enables RFFIF interrupt (RIS[2] register). Since the INTEN[2] register is set to 0 or the TFT is set to a value greater than 0, if the

RFFIF interrupt has been masked, it is recommended to implement a timing program for the periodic reading of STS1 register. The reading operation of the STS1 register with Bit 3 (RFNEF) set to 1 must be considered by the software as equivalent to the RFFIF interrupt enabled.

（5） The software can read bytes (Bit 7:0) from DATA register.

（6） Another master can retain the I2C mainline by issuing the restart (RSTAEN) condition or release the mainline by sending a stop (STOP) condition.

Slave transmission operation of bulk transmission

In the standard I2C protocol, all transmissions are single-byte transmission, and responds to the remote master read request by writing a byte to the FIFO of the slave. When the slave (slave transmitter) responds to a read request from the remote master (master receiver), at least one entry should be put in the TXFIFO of the slave transmitter. I2C is used to process more data in TXFIFO, so that the subsequent read requests can get more data without causing any interrupt.

Finally, this eliminates the possibility of occurrence of major delay between data interrupts every time only one entry is placed in TXFIFO.

This mode occurs only when I2C acts as a slave transmitter. If the remote master confirms the data transmitted by the slave and there is no data in the TXFIFO of the slave, I2C will keep the row in low bit while sending the read request interrupt (RRIF) and wait for the data to be written into the TXFIFO before the data is transmitted to the remote master.

If the RRIF interrupt is masked, since Bit 5 (RRIE) of the ISTS register is set to 0, it is recommended to use a timing program to activate periodic reading of the RIS register. The read operation of RIS with Bit 5 (RRIF) restored to 1 must be regarded as equivalent read operation to the RRIF interrupt mentioned in this section.

RRIF interrupts are triggered when reading requests. Like interrupts, they must be cleared when exiting the interrupt service processing routine (ISR). ISR allows you to write one or more bytes to TXFIFO. During transmission of these bytes to the master, if the master confirms the last byte, the slave must issue a RRIF again, which is because the master is requesting more data.

If knows in advance that the remote master is requesting an n-byte data packet, when another master addresses I2C and requests data, TXFIFO can be written in n bytes, and the remote master receives it as a continuous data stream. For example, the I2C slave will continue to send data to the remote master as long as the remote master is confirming the transmitted data and there is data available in TXFIFO. There is no need to keep the SCL row in low bit or issue a RRIF again.

If the remote master wants to receive n bytes from I2C, but writes the bytes greater than n into TXFIFO, when the slave completes sending the requested n bytes, the master will clear TXFIFO and ignore any redundant bytes.

I2C generates a transfer abort (TAIF) event to indicate the clearing of TXFIFO in this case. When ACK/NACK is needed, if NACK is received, the remote master will have all the required data. At this time, the state machine of the slave issues a flag to clear the remaining data in TXFIFO. This flag is transmitted to the processor bus clock domain where the FIFO is located, and the content of TXFIFO is cleared.

22.2.5.2 Master mode operation

Initial Configuration

As the master of I2C, the destination address and address format can be changed without disabling I2C. The slave requires that the component be disabled before making any changes to the addressing.

（1） Write 0 to Bit 0 of CTRL1 register to disable I2C.

（2） When the device is slave (Bit 3), write to CTRL0 register to set the maximum speed mode (Bit 2:1) supported by slave operation, and specify whether I2C starts transmission in 7/10-bit addressing mode.

（3） Write the address of the I2C device to be addressed to the TARADDR register. It also indicates I2C will execute the general call or start byte command. The speed required for the transmission (7-bit or 10-bit address) initiated by the I2C master controller is controlled by the MAM bit field (Bit 12).

（4） Apply only to transmission in high-speed mode. Write the main code required by I2C to HSMC register. The main code is defined by the programmer.

（5） Write 1 to Bit 0 of CTRL1 register to enable I2C.

（6） Now write the transmission direction and data to be transmitted to the DATA register. If the DATA register is written before I2C is enabled, the data and commands will be lost with the clearing of buffer when I2C is not enabled.

22.2.6 Spike Suppression

I2C contains programmable peak suppression logic, which meets the requirements of I2C bus specification for SS/FS and HS modes.

This logic is based on the counter that monitors the input signals (SCL and SDA). Each signal has a separate counter (SCL and SDA). The number of IC_CLK cycles can be programmed by the user, and the frequency of IC_CLK and related peak length specifications should be considered in the calculation.

Each counter will be activated whenever the input signal value is changed. One of the following situations will occur, depending on the behavior of the input signal:

The input signal remains unchanged until the counter reaches the count limit. When this situation happens, the internal version of the signals will be updated with the input value and the counter will be reset and stopped. The counter will not restart until a new change to the input signal is detected.

Before the counter reaches the count limit, the input signal will change again. When this situation happens, the counter will be reset and stopped, but the internal version of the signal will not be updated. The counter remains in stop state until a new change on the input signal is detected.

The width of these registers is 8 bits, which can be read and written through APB interface. However, they can be written only when I2C is disabled. The minimum value that can be programmed into these registers is 1. If you attempt to write a value less than 1, 1 will be written.

22.3 Register Address Mapping

Table 94 I2C3 Register Address Mapping

Register name	Description	Offset address
CTRL0	Control register	0x00
TARADDR	Destination address register	0x04
SLAADDR	Slave address register	0x08
HSMC	Master address register	0x0C
DATA	Data command register	0x10
SSCHC	SCL high-bit counter register of standard-speed I2C clock	0x14
SSCLC	SCL low-bit counter register of standard-speed I2C clock	0x18
FSCHC	SCL high-bit counter register of I2C clock of fast-speed mode or fast-speed plus mode	0x1C
FSCLC	SCL low-bit counter register of I2C clock of fast-speed mode or fast-speed plus mode	0x20
HSCHC	SCL high-bit counter register of high-speed mode I2C clock	0x24
HSCLC	SCL low-bit counter register of high-speed mode I2C clock	0x28
ISTS	Interrupt state register	0x2C
INTEN	Interrupt enable register	0x30
RIS	Original interrupt state register	0x34
RFT	Transmit FIFO watermark register	0x38
TFT	Receive FIFO watermark register	0x3C
ICF	Clear combined and single interrupt register	0x40

Register name	Description	Offset address
RFUIC	Clear RFUIT interrupt register	0x44
RFOIC	Clear RFOIF interrupt register	0x48
TFOIC	Clear TFOIF interrupt register	0x4C
RRIC	Clear RRIF interrupt register	0x50
TAIC	Clear TAIF interrupt register	0x54
RDIC	Clear RDIF interrupt register	0x58
AIC	Clear ACTIF interrupt register	0x5C
STPDIC	Clear STPDIF interrupt register	0x60
STADIC	Clear STADIF interrupt register	0x64
GCIC	Clear GCIF interrupt register	0x68
CTRL1	Start the register	0x6C
STS1	State register	0x70
TFL	Transmit FIFO stock register	0x74
RFL	Receive FIFO stock register	0x78
SDAHOLD	SDA hold time register	0x7C
TAS	Transfer abort source register	0x80
SDNO	Generate slave data NACK register	0x84
DMACTRL	DMA control register	0x88
DTDL	DMA transmitting data level register	0x8C
DRDL	DMA receive data level register	0x90
SDADLY	SDA setup register	0x94
GCA	ACK general call register	0x98
STS2	Enable state register	0x9C
LSSSL	SS, FS or FM+ spike suppression limit register	0xA0
HSSSL	HS spike suppression limit register	0xA4

22.4 Register Functional Description

22.4.1 I2C control register (CTRL0)

Offset address: 0x00

Reset value: 0x3E

Field	Name	R/W	Description
0	MST	R/W	Master mode controller Whether this bit controls 2C master 1: Enable master mode 0: Disable master mode
2:1	SPEED	R/W	I2C operating speed These bits control the speed of I2C operation; it can be set only when I2C is operated in master mode. Hardware can prevent illegal values written by software. These bits must be properly programmed for the slave mode as they are used to capture the

Field	Name	R/W	Description
			correct narrow band filter value according to the speed mode. The programming value of this register can only be between 1 and 3; otherwise, the hardware will update the value of this register to 3. 01: Standard mode (100kbit/s) 10: Fast-speed mode (<=400kbit/s) or fast-speed mode+ (<=1000Kbit/s) 11: High-speed mode (3.4Mbit/s)
3	SAM	R/W	Select the addressing mode 0: 7-bit addressing mode 1: 10-bit addressing mode
4	Reserved		
5	RSTAEN	R/W	Transmitting the repeat start bit allowed 0: Transmitting repeat start bit not supported. 1: Transmitting repeat start bit supported. Confirm whether RSTAEN can be sent when it acts as a master. Some older slaves do not support processing RSTAEN; however, RSTAEN is used in some I2C operations. When restart is disabled, the main program is disabled to perform the following functions: transmit a start byte (START BYTE) Perform any high-speed mode operation High-speed mode operation Perform direction changes in combination mode Perform read operation with 10-bit address By replacing RSTAEN with stop and following START, discompose the operation into multiple I2C transmissions. If the above operation is performed, Bit 6 (TAIF) of RIS register will be set.
6	SLADIS	R/W	Enable the slave mode 0: Slave mode enabled 1: Slave mode disabled This bit controls whether I2C has disabled its slave. After the reset is applied, you can choose to enable or disable the slave, which means that the software does not need to configure the slave. By default, the slave is always enabled (also in reset state). If you need to disable it after reset, set this bit to 1. If this bit is set (the slave is disabled), I2C will only run as the master and will not perform any operation that requires the slave.
7	DSA	R/W	In slave mode: 1: (ENABLED): As long as it is addressed, the slave will issue STPDIF interrupt 0: (DISABLED): The slave always issues STPDIF interrupt Note: During general call addressing, if DSA=1b1, this slave will not issue STPDIF interrupt, even if the slave responds to the general call addressing by generating ACK. STPDIF interrupt is generated only when the transmission address matches the slave address (SAR).
8	TFEIC	R/W	This bit controls the generation of TFEIF interrupts, as described in the RIS register. 1 (ENABLED): Control generation of TFEIF interrupt 0 (DISABLED): Default behavior of TFEIF interrupt
9	RFFIE	R	This bit controls whether I2C maintains the bus when RXFIFO is full 1: (ENABLED): Maintain the bus when RX_FIFO is full 0: (DISABLED): Overrun when RX_FIFO is full

Field	Name	R/W	Description
10	DSMA	R	In master mode: 1: (ENABLED): Enable STPDIF interrupt only when the master is running 0: (DISABLED): Enable STPDIF interrupt regardless of whether the master is running
31:11	Reserved		

22.4.2 I2C destination address register (TARADDR)

Offset address: 0x4

Reset value: 0x1055

Field	Name	R/W	Description
8:0	ADDR	R/W	This is the destination address of any master transaction. In GENERAL_CALL, these bits will be ignored. To generate START_BYTE, the CPU only needs to write these bits once. If ADDR and SAR are the same, they can transmit to themselves, but FIFOs are shared between master and slave, and they cannot transmit to themselves. Only support simplex mode, and do not support duplex mode. The master cannot transmit information to itself; it can only transmit to one slave.
10:9	STA	R/W	If the GCEN bit is set to 1, it indicates I2C performs the broadcast or start byte command. 1: START_BYTE: start byte transmission 0: GENERAL_CALL: broadcast byte transmission
11	GCEN	R/W	This bit indicates whether the software executes the device ID, or general call, or start byte command. 1: (ENABLED): Enable the program transmitted by GENERAL_CALL or START_BYTE 0: (DISABLED): Disable the program transmitted by GENERAL_CALL or START_BYTE
12	MAM	R/W	When I2C is the master, this bit is configured to start its transmission in 7-bit or 10-bit addressing mode. 1: (ADDR_10BITS): Addressing 10-bit transmission format 0: (ADDR_7BITS): Addressing 7-bit transmission format
31:13	Reserved		

22.4.3 I2C slave address register (SLAADDR)

Offset address: 0x8

Reset value: 0x055

Field	Name	R/W	Description
9:0	SAR	R/W	When I2C runs as a slave, SLAADDR will save the slave address. For 7-bit addressing, only use SLAADDR[6:0]. This register can write only when I2C interface is disabled, corresponding to CTRL1[0] register set to 0. Writing is invalid in other cases.
31:10	Reserved		

22.4.4 Main address register (HSMC)

Offset address: 0xC

Reset value: 0x7

Field	Name	R/W	Description
2:0	HSMC	R/W	This bit field saves the value of I2C HS mode main code. The HS mode master code is a reserved 8-bit code (00001xxx) and is not used for slave addressing or other purposes. Each masterhas its own unique main code; in the same I2C bus system, up to 8 high-speed mode masters can be provided. The valid values range from 0 to 7. This register disappears and returns 0, and becomes read-only. This register can write only when I2C interface is disabled, corresponding to CTRL1[0] register set to 0. Writing is invalid in other cases.
31:3	Reserved		

22.4.5 Data command register (DATA)

Offset address: 0x10

Reset value: 0x00

Field	Name	R/W	Description
7:0	DATA	R/W	This register contains the data to be transmitted or received on the I2C bus. Reading this bit when writing is invalid. When reading this bit, the value returned by these bits is the data received on the I2C interface.
8	CMD	W	This bit controls to perform read or write operations. When I2C acts as a slave, this bit will not control the direction. When it is the master, it controls the direction. When you enter a command in TXFIFO, this bit will distinguish write and read commands. In slave receiving mode, this bit is useless. In slave transmitting mode, "0" means to transmit the data in DATA. When writing this bit, you should remember that if you attempt to perform a read operation after transmitting a general call command, a TAIF interrupt (Bit 6 of the RIS register) will be caused, unless Bit 11 (GCEN) of the TARADDR register has been cleared. If "1" is written to this bit after receiving the RRIF interrupt, a TAIF interrupt will occur. 0: Sequence data byte receiving, Main write command 1: Main read command
9	STOP	R/W	This bit controls stopping transmitting or receiving of data.
31:10	Reserved		

22.4.6 SCL high-bit counter register of standard-speed I2C clock (SSCHC)

Offset address: 0x14

Reset value: 0x190

Field	Name	R/W	Description
13:0	CNT	R/W	This register must be set before any I2C bus transaction to ensure the correct input/output (I/O) time. This register sets the high 16 bits of the SCL clock counter in standard speed mode. For more information, please see "IC_CLK Frequency Configuration". This register can only write when I2C interface is disabled (CTRL1[0] register is 0). Writing is invalid in other cases. The minimum effective value is 6; The hardware can prevent writing values less than this value, and if you attempt to set a value, the set value of the result is 6. For the design of APB_DATA_WIDTH=16, the programming sequence is very important for ensuring correct operation of I2C. First program the lower byte. Then program the upper byte.
31:14	Reserved		

22.4.7 SCL low-bit counter register of standard-speed I2C clock (SSCLC)

Offset address: 0x18
Reset value: 0x1D6

Field	Name	R/W	Description
13:0	CNT	R/W	This register must be set before any I2C bus transaction to ensure the correct input/output (I/O) time. This register sets the low 16 bits of the SCL clock counter in standard speed mode. For more information, please refer to "IC_CLK Frequency Configuration". This register can only write when I2C interface is disabled (CTRL1[0] register is 0). Writing is invalid in other cases. The minimum effective value is 8; the hardware can prevent writing values less than this value, and if you attempt to set a value, the set value of the result is 8. For the design of APB_DATA_WIDTH=16, the programming sequence is very important for ensuring correct operation of I2C. First program the lower byte, and then program the upper byte.
31:14	Reserved		

22.4.8 SCL high-bit counter register of I2C clock of fast-speed mode or fast-speed plus mode (FSCHC)

Offset address: 0x1C
Reset value: 0x3C

Field	Name	R/W	Description
13:0	CNT	R/W	This register must be set before any I2C bus transaction to ensure the correct input/output (I/O) time. This register sets the high 16 bits of the SCL clock counter in fast speed mode or fast speed plus mode. It is used to transmit the main code and START BYTE or General CALL in fast speeed mode. For more information, please see "IC_CLK Frequency Configuration". This register is 0 when reset, and becomes read-only. This register can only write when I2C interface is disabled (CTRL1[0] register is 0). The minimum effective value is 6; the hardware can prevent writing values less than this value, and if you attempt to set a value, it is set to 6. For the design of APB_DATA_WIDTH=16, the programming sequence is very important for ensuring correct operation of I2C. First program the lower byte. Then program the upper byte.
31:14	Reserved		

22.4.9 SCL low-bit counter register of I2C clock of fast-speed mode or fast-speed plus mode (FSCLC)

Offset address: 0x20
Reset value: 0x82

Field	Name	R/W	Description
13:0	CNT	R/W	This register must be set before any I2C bus transaction to ensure the correct input/output (I/O) time. This register sets the high 16 bits of the SCL clock counter in fast speed mode or fast speed plus mode. It is used to transmit the main code and START BYTE or General CALL in fast speeed mode. For more information, please see "IC_CLK Frequency Configuration". This register is 0 when reset, and becomes read-only. This register can only write when I2C interface is disabled (CTRL1[0] register is 0), and write is ineffective in other cases. The minimum effective value is 8; the hardware can prevent writing values less than this value, and if you attempt to set a value, the set value of the result is 8. For the design of APB_DATA_WIDTH=16, the programming sequence is very important for ensuring correct operation of I2C. First program the lower byte. Then program the upper byte. If the value is less than 8, the count value will be changed to 8.

Field	Name	R/W	Description
31:14	Reserved		

22.4.10 SCL high-bit counter register of high-speed mode I2C clock (HSCHC)

Offset address: 0x24

Reset value: 0x6

Field	Name	R/W	Description
13:0	CNT	R/W	This register must be set before any I2C bus transaction to ensure the correct input/output (I/O) time. This register sets the high 16 bits of the SCL clock counter in high-speed mode. Please see "IC_CLK Frequency Configuration". Setting of this register depends on the bus. It is 60ns when 100pF is loaded; 120ns when 400pF is loaded. This register is 0 when reset, and becomes read-only. This register can only write when I2C interface is disabled (CTRL1[0] register is 0), and write is ineffective in other cases. The minimum effective value is 6; the hardware can prevent writing values less than this value, and if you attempt to set a value, the final set value is 6. For the design of APB_DATA_WIDTH=16, the programming sequence is very important for ensuring correct operation of I2C. First program the lower byte. Then program the upper byte.
31:14	Reserved		

22.4.11 SCL low-bit counter register of high-speed mode I2C clock (HSCLC)

Offset address: 0x28

Reset value: 0x10

Field	Name	R/W	Description
13:0	CNT	R/W	This register must be set before any I2C bus transaction to ensure the correct input/output (I/O) time. This register sets the low 16 bits of the SCL clock counter in high-speed mode. For more information, please see "IC_CLK Frequency Configuration". Setting of this register depends on the bus. It is 160ns when 100pF is loaded; 320ns when 400pF is loaded. This register is 0 when reset, and becomes read-only. This register can only write when I2C interface is disabled (CTRL1[0] register is 0), and write is ineffective in other cases. The minimum effective value is 8; the hardware can prevent writing values less than this value, and if you attempt to set a value, the final set value is 8. For the design of APB_DATA_WIDTH=16, the programming sequence is very important for ensuring correct operation of I2C. First program the lower byte. Then program the upper byte. If the value is less than 8, the count value will be changed to 8.
31:14	Reserved		

22.4.12 I2C interrupt state register (INTSTS)

Offset address: 0x2C

Reset value: 0x00

Field	Name	R/W	Description
0	RFUIF	R	For detailed description of RFUIF bit, please see RIS. 1: RFUIF interrupt in active state 0: RFUIF interrupt in inactive state
1	RFOIF	R	For detailed description of RFOIF bit, please see RIS. 1: RFOIF interrupt in active state

Field	Name	R/W	Description
			0: RFOIF interrupt in inactive state
2	RFFIF	R	For detailed description of RFFIF bit, please see RIS. 1: RFFIF interrupt in active state 0: RFFIF interrupt in inactive state
3	TFOIF	R	For detailed description of TFOIF bit, please see RIS. 1: TFOIF interrupt in active state 0: TFOIF interrupt in inactive state
4	TFEIF	R	For detailed description of TFEIF bit, please see RIS. 1: TFEIF interrupt in active state 0: TFEIF interrupt in inactive state
5	RRIF	R	For detailed description of RRIF bit, please see RIS. 1: RRIF interrupt in active state 0: RRIF interrupt in inactive state
6	TAIF	R	For detailed description of TAIF bit, please see RIS. 1: TAIF interrupt in active state 0: TAIF interrupt in inactive state
7	RDIF	R	For detailed description of RDIF bit, please see RIS. 1: RDIF interrupt in active state 0: RDIF interrupt in inactive state
8	ACTIF	R	For detailed description of R_ACTIF bit, please see RIS. 1: R_ACTIF interrupt in active state 0: R_ACTIF interrupt in inactive state
9	STPDIF	R	For detailed description of STPDIF bit, please see RIS. 1: STPDIF interrupt in active state 0: STPDIF interrupt in inactive state
10	STADIF	R	For detailed description of STADIF bit, please see RIS. 1: STADIF interrupt in active state 0: STADIF interrupt in inactive state
11	GCIF	R	For detailed description of GCIF bit, please see RIS. 1: GCIF interrupt in active state 0: GCIF interrupt in inactive state
12	RSTADIF	R	For detailed description of RSTADIF bit, please see RIS. 1: RSTADIF interrupt in active state 0: RSTADIF interrupt in inactive state
13	MOHIF	R	For detailed description of MOHIF bit, please see RIS. 1: MOHIF interrupt in active state 0: MOHIF interrupt in inactive state
31:14	Reserved		

22.4.13 I2C interrupt enable register (INTEN)

Offset address: 0x30

Reset value: 0x8FF

Field	Name	R/W	Description
0	RFUIE	R/W	This bit enables RFUIF interrupt in the ISTS register. 1: Enable 0: Disable
1	RFOIE	R/W	This bit enables RFOIF interrupt in the ISTS register. 1: Enable 0: Disable

Field	Name	R/W	Description
2	RFFIE	R/W	This bit enables RFFIF interrupt in the ISTS register. 1: Enable 0: Disable
3	TFOIE	R/W	This bit enables TFOIF interrupt in the ISTS register. 1: Enable 0: Disable
4	TFEIE	R/W	This bit enables TFEIF interrupt in the ISTS register. 1: Enable 0: Disable
5	RRIE	R/W	This bit enables RRIF interrupt in the ISTS register. 1: Enable 0: Disable
6	TAIE	R/W	This bit enables TAIF interrupt in the ISTS register. 1: Enable 0: Disable
7	RDIE	R/W	This bit enables RDIF interrupt in the ISTS register. 1: Enable 0: Disable
8	ACTIE	R/W	This bit enables R_ACTIF interrupt in the ISTS register. 1: Enable 0: Disable
9	STPDIE	R/W	This bit enables STPDIF interrupt in the ISTS register. 1: Enable 0: Disable
10	STADIE	R/W	This bit enables STADIF interrupt in the ISTS register. 1: Enable 0: Disable
11	GCIE	R/W	This bit enables GCIF interrupt in the ISTS register. 1: Enable 0: Disable
31:12	Reserved		

22.4.14 I2C original interrupt state register (RIS)

Offset address: 0x34

Reset value: 0x00

Field	Name	R/W	Description
0	RFUIF	R	Set when the receive buffer is empty and the processor attempts to read the receive buffer from the DATA register. If the module is disabled (CTRL1[0]=0), this bit holds its level until the master state machine or slave state machine enters the idle state. When I2CEN becomes 0, this interrupt will be cleared. 1: RFUIF interrupt in active state 0: RFUIF interrupt in inactive state
1	RFOIF	R	If the receive buffer has been fully filled, it is set to the IC_RX_BUFFER_DEPTH and receives another byte from the external I2C device. I2C confirms this, but any data bytes received after FIFO is full will be lost. If the module is disabled (CTRL1[0]=0), this bit holds its level until the master state machine or slave state machine enters the idle state. When I2CEN becomes 0, this interrupt will be cleared.

Field	Name	R/W	Description
			1: RFOIF interrupt in active state 0: RFOIF interrupt in inactive state
2	RFFIF	R	Set when the receive buffer reaches or exceeds the RFT watermark in the RFT register. When the buffer level is below the watermark, the hardware automatically clears it. If the module is disabled (CTRL1[0] = 0), refresh the RX FIFO and save it in the reset; therefore, the RX FIFO is not full. Once CTRL1 Bit 0 is programmed to 0, this bit will be cleared regardless of the activity to continue. 1: RFFIF interrupt in active state 0: RFFIF interrupt in inactive state
3	TFOIF	R	If the transmit buffer is filled to 16 and the processor attempts to issue another I2C command by writing to the DATA register, it is set during transmission. When the module is disabled, this bit holds its level until the master or slave state machine enters the idle state. When I2CEN becomes 0, this interrupt will be cleared. 1: TFOIF interrupt in active state 0: TFOIF interrupt in inactive state
4	TFEIF	R	According to the TFEIC selection in CTRL0 register, the behavior of TFEIF interrupt state is different. When TFEIC=0: this bit is set to 1 when the transmit buffer is at or below the watermark set in the TFT register. When TFEIC=1: when the transmit buffer is at or below the watermark value set in the TFT register, this bit is set to 1, and the transmission of address/data from the internal shift register of the latest pop-up command is completed. When the buffer level exceeds the watermark, the hardware automatically clears it. When CTRL1[0] is set to 0, TXFIFO is refreshed and saved in reset. The TXFIFO here appears to have no data, so this bit is set to 1, provided it is active in the master or slave state machine. When there is no more activity, and I2CEN=0, this bit is set to 0. 1: TFEIF interrupt in active state 0: TFEIF interrupt in inactive state
5	RRIF	R	This bit is set to 1 when I2C acts as a slave server and another I2C master attempts to read data from I2C. I2C keeps the I2C bus in the waiting state (SCL=0) until the interrupt is served, which means that the slave machine has been addressed by the remote master requesting data transmission. The processor must respond to the interrupt and write the requested data to the DATA register. This bit is set to 0 after the processor reads IC_CLRRIF register. 1: RRIF interrupt in active state 0: RRIF interrupt in inactive state
6	TAIF	R	This bit indicates that if I2C acts as an I2C transmitter, it cannot complete the expected operation of transmit FIFO content. This situation can occur on either the I2C master or the I2C slave, which is called "transfer abort". When this bit is set to 1, the TAS register indicates the cause of the transfer abort. Note: I2C only refreshes/resets/clears TX_FIFO when any event tracked by the TAS register causes the transfer abort. Before reading the register IC_CLTAIF, TXFIFO keeps this refresh state. After performing this read, TXFIFO can receive more data bytes from APB interface. 1: TAIF interrupt in active state

Field	Name	R/W	Description
			0: TAIF interrupt in inactive state
7	RDIF	R	This bit masks the RDIF interrupt in the ISTS register. 1: RDIF interrupt unmasked 0: RDIF interrupt masked
8	ACTIF	R	This bit captures I2C activity and remains set until it is cleared. There are four ways to clear it: I2C is disabled; Read AIC register; Read ICF register; System reset. Once this bit is set, it remains set unless one of these four ways is used to clear it. Even if the I2C module is idle, this bit remains set until cleared, indicating that there is activity on the bus. 1: RAW_INTR_ACTIF interrupt in active state 0: RAW_INTR_ACTIF interrupt in inactive state When I2C acts as a slave transmitter, if the master does not confirm the bytes transmitted, set the bit to 1. This occurs in the last byte transmitted, indicating that the transmission has been completed. 1: RDIF interrupt in active state 0: RDIF interrupt in inactive state
9	STPDIF	R	Indicate whether a stop condition has occurred on the I2C interface, regardless of the I2C running in slave or master mode. In slave mode: If CTRL0 register DSA=1b1, the STPDIF interrupt is only issued when the slave machine is addressed. Note: During general call addressing, if DSA=1b1, this slave will not issue an STPDIF interrupt, even if the slave responds to the general call addressing by generating an ACK. STPDIF interrupt is generated only when the transmission address matches the slave address (SAR). If CTRL0 register DSA=1b0, the STOP DET interrupt will be issued regardless of whether it is addressed. In master mode: If CTRL0 register DSMA=1b1, the STPDIF interrupt will be issued only when the master is active. If CTRL0 register DSMA=1b0, the STPDIF interrupt will be issued regardless of whether the master is active. 1: STPDIF interrupt in active state 0: STPDIF interrupt in inactive state
10	STADIF	R	Indicate whether a start (START) or restart (RSTAEN) condition has occurred on the I2C interface, regardless of the I2C running in slave or master mode. 1: STADIF interrupt in active state 0: STADIF interrupt in inactive state
11	GCIF	R	Set only when a general call addressing is received and the address is confirmed. It remains set until it is cleared by disabling I2C or when the CPU reads Bit 0 of the GCIC register. I2C stores the received data in the Rx buffer. 1: GCIF interrupt in active state 0: GCIF interrupt in inactive state
12	RSTADIF	R	Indicate whether a restart (RSTAEN) condition occurs on the I2C interface when I2C is operating in slave mode and addressing the slave. Enable only when

Field	Name	R/W	Description
			IC_SLV_RSTADIF_EN=1. Note: However, in high-speed mode or during initial byte transmission, restart before the address field according to I2C protocol. In this case, when a restart is issued, the slave is not addressing slave, so I2C will not generate RSTADIF interrupt. 1: RSTADIF interrupt in active state 0: RSTADIF interrupt in inactive state
13	MOHIF	R	Indicate whether master holds bus, and whether TXFIFO is empty. Enable only when I2C_DYNAMIC_TAR_UPDATE=1 and IC_EMPTYFIFO_HOLD_MASTER_EN=1. 1: MOHIF interrupt is active 0: MOHIF interrupt is inactive
31:14	Reserved		

22.4.15 I2C receiver FIFO watermark register (RFT)

Offset address: 0x38

Reset value: 0x00

Field	Name	R/W	Description
7:0	RFT	R/W	The receive FIFO watermark controls the data bits that trigger the RFFIF interrupt (Bit 2 in RIS register). If it is greater than this value, the interrupt will be triggered. The valid range is 0-255, but the hardware does not allow setting this value to be greater than the buffer depth. If you try to do this, the actual value will be the maximum depth of the buffer. The value 0 sets the watermark of 1 bit, and the value 255 sets the watermark of 256 bits.
31:8	Reserved		

22.4.16 I2C transmit FIFO watermark register (TFT)

Offset address: 0x3C

Reset value: 0x00

Field	Name	R/W	Description
7:0	TFT	R/W	The transmit FIFO watermark controls trigger TFEIF interrupt (Bit 4 in RIS register); the interrupt will be triggered if the value is less than this value. The valid range is 0-255, and the additional restriction is that it cannot be set to be greater than the buffer depth. If you try to do this, the actual value will be the maximum depth of the buffer. The value 0 sets the watermark of 0 bit, and the value 255 sets the watermark of 255 bits.
31:8	Reserved		

22.4.17 Clear combined and single interrupt register (ICF)

Offset address: 0x40

Reset value: 0x00

Field	Name	R/W	Description
0	ICF	R	Read this register to clear the combined interrupt, all separate interrupts, and TAS registers. This bit does not clear the interrupts that can be cleared by hardware, but clears the interrupts that can be cleared by software. For clearing TAS exception, refer to Bit 9 of the TAS register.
31:1	Reserved		

22.4.18 Clear RFUIT interrupt register (RFUIC)

Offset address: 0x44

Reset value: 0x00

Field	Name	R/W	Description
0	RFUIC	R	Read this register to clear the RFUIF interrupt (Bit 0) of RIS register.
31:1	Reserved		

22.4.19 Clear RFOIF interrupt register (RFOIC)

Offset address: 0x48

Reset value: 0x00

Field	Name	R/W	Description
0	RFOIC	R	Read this register to clear the RFOIF interrupt (Bit 1) of RIS register.
31:1	Reserved		

22.4.20 Clear TFOIF interrupt register (TFOIC)

Offset address: 0x4C

Reset value: 0x00

Field	Name	R/W	Description
0	TFOIC	R	Read this register to clear the TFOIF interrupt (Bit 3) of RIS register.
31:1	Reserved		

22.4.21 Clear RRIF interrupt register (RRIC)

Offset address: 0x50

Reset value: 0x00

Field	Name	R/W	Description
0	RRIC	R	Read this register to clear the RRIF interrupt (Bit 5) of RIS register
31:1	Reserved		

22.4.22 Clear TAIF interrupt register (TAIC)

Offset address: 0x54

Reset value: 0x00

Field	Name	R/W	Description
0	TAIC	R	Read this register to clear the TAIF interrupt (Bit 6) of RIS register and TAS register. This can also relese the TXFIFO from the refresh/reset state, thus allowing more writes to the TXFIFO. For clearing TAS exception, refer to MSTDIS bit of the TAS register.
31:1	Reserved		

22.4.23 Clear RDIF interrupt register (RDIC)

Offset address: 0x58

Reset value: 0x00

Field	Name	R/W	Description
0	RDIC	R	Read this register to clear the RDIF interrupt (Bit 7) of RIS register.
31:1	Reserved		

22.4.24 Clear ACTIF interrupt register (AIC)

Offset address: 0x5C

Reset value: 0x00

Field	Name	R/W	Description
0	AIC	R	If I2C is no longer active, read this register to clear the ACTIF interrupt. If the I2C module is still active on the bus, continue to set the ACTIF interrupt bit. If the module is disabled and there is no other activity on the bus, the hardware will automatically clear this interrupt bit. Read this register to obtain the value of ACTIF interrupt state (Bit 8) of RIS register.
31:1	Reserved		

22.4.25 Clear STPDIF interrupt register (STPDIC)

Offset address: 0x60

Reset value: 0x00

Field	Name	R/W	Description
0	STPDIC	R	Read this register to clear the STPDIF interrupt (Bit 9) of RIS register.
31:1	Reserved		

22.4.26 Clear STADIF interrupt register (STADIC)

Offset address: 0x64

Reset value: 0x00

Field	Name	R/W	Description
0	STADIC	R	Read this register to clear the STADIF interrupt (Bit 10) of RIS register.
31:1	Reserved		

22.4.27 Clear GCIF interrupt register (GCIC)

Offset address: 0x68

Reset value: 0x00

Field	Name	R/W	Description
0	GCIC	R	Read this register to clear the GCIF interrupt (Bit 11) of RIS register.
31:1	Reserved		

22.4.28 I2C start register (CTRL1)

Offset address: 0x6c

Reset value: 0x00

Field	Name	R/W	Description
0	I2CEN	R/W	Control to enable I2C or not 0: Disable I2C (TX and RX FIFO remain in erased state) 1: Enable I2C When I2C is active, the software can disable it. However, it must be ensured that I2C is correctly disabled. A program is recommended in the "disable I2C" description. When I2C is disabled, the following may occur: TXFIFO and RXFIFO refresh

Field	Name	R/W	Description
			The state bit in the ISTS register is still active until I2C enters the idle (IDLE) state. If the module is transmitting, it will stop and delete the content of the transmit buffer after the current transmission is completed. If the module is receiving, I2C will stop the current transmission at the end of the current byte and will not confirm the transmission.
1	ABR	R/W	After setting, the controller enables the transfer abort. 0: Abort is not started or has been completed 1: Abort operation in progress By setting this bit, the software can abort I2C transfer in master mode. The software can only set this bit when enabling (I2CEN) has been set; otherwise, the controller will ignore any write to the abort (ABR) bit. Once set, the software cannot clear the abort bit. In response to the abort, the controller issues a STOP and refreshes the TXFIFO after the current transmission is completed, and then sets the TX_ABORT interrupt after the abort operation . The abort bit is automatically cleared after the abort operation.
2	TCB	R/W	1: Even if TXFIFO has data for transmission, it will also block transmission of the data on I2C bus. 0: When the first data in TXFIFO is available, the data will be automatically transmitted on I2C bus. Note: When TXFIFO is empty (STS1[2]==1) and the master is in idle state (STS1[5]==0), set TCB bit and the master will stop executing the command. Command in TXFIFO will not be executed before TCB bit is set.
31:3	Reserved		

22.4.29 I2C state register (STS1)

Offset address: 0x70

Reset value: 0x06

This is a read-only register, used to indicate current transmission state and FIFO state. The state register can read at any time. There is no field request interrupt in this register. When I2C is disabled by writing 0 to Bit 0 of CTRL1 register: Bit 1 and bit 2 are set to 0, Bit 3 and Bit 10 are set to 0. When the master or slave state machine is idle and I2CEN=0: Bit 5 and Bit 6 are set to 0

Field	Name	R/W	Description
0	ACTF	R	I2C active state 1: I2C in active state 0: I2C in idle state
1	TFNFF	R	Transmit FIFO not full. Set when the transmit FIFO contains one or more idle positions; clear when FIFO is full. 1: TXFIFO not full 0: TXFIFO full
2	TFEF	R	Transmit FIFO is completely empty. When the transmit FIFO is completely empty. set this bit. When it contains one or more valid entries, clear this field. This bit field does not request interrut. 1: TXFIFO empty 0: TXFIFO not empty
3	RFNEF	R	Receive FIFO is not empty. Set this bit when the receive FIFO contains one or more entries; clear it when the receive FIFO is empty. 1: Rx FIFO not empty 0: Rx FIFO empty

Field	Name	R/W	Description
4	RFFF	R	Receive FIFO is completely full. Set this field when Receive FIFO is completely full. Clear this field when receive FIFO contains one or more empty positions. 1: RXFIFO full 0: RXFIFO not full
5	MAF	R	Master in active state. Set this bit when the master is not in idle state. 1: The master is not idle 0: The master is idle
6	SAF	R	Slave in active state. Set this bit when the slave is not in idle (IDLE) state. 1: The slave is not idle 0: The slave is idle
31:7	Reserved		

22.4.30 I2C transmit FIFO stock register (TFL)

Offset address: 0x74
Reset value: 0x00

Field	Name	R/W	Description
3:0	TFL	R	Data volume in transmit FIFO Include the quantity of valid data in transmit FIFO.
31:4	Reserved		

22.4.31 I2C receive FIFO stock register (RFL)

Offset address: 0x78
Reset value: 0x00

Field	Name	R/W	Description
3:0	RFL	R	Data volume in receive FIFO Include the quantity of valid data in receive FIFO.
31:4	Reserved		

22.4.32 I2C SDA hold time register (SDAHOLD)

Offset address: 0x7C
Reset value: 0x01

Field	Name	R/W	Description
15:0	TXHOLD	R/W	When I2C is used as a transmitter, set the required SDA hold time with IC_CLK cycle as the unit.
23:16	RXHOLD	R/W	When I2C is used as a receiver, set the required SDA hold time with IC_CLK cycle as the unit.
31:24	Reserved		

22.4.33 I2C transmitter abort source register (TAS)

Offset address: 0x80
Reset value: 0x00

Field	Name	R/W	Description
0	AD7NA	R	This field indicates that the Master is in 7-bit addressing mode and the sent address is not confirmed by any slave. 1: This abort is generated by NOACK of 7-bit address 0: This abort not generated

Field	Name	R/W	Description
1	AD10NA1	R	This field indicates that the Master is in 10-bit addressing mode, and no slave confirms the first byte of the 10-bit address. 1: No first byte of ACKed 10-bit address of any slave 0: There is slave response
2	AD10NA2	R	This field indicates that the Master is in 10-bit addressing mode and the second address byte of this 10-bit addressing is not confirmed by any slave. 1: No byte 2 of ACKed 10-bit address of any slave 0: There is response.
3	TDNA	R	This field indicates only master mode field. When the master receives the address acknowledgment, and it transmit the data byte after the address, it indicates that it has not received the acknowledgment from the remote slave. 1: The addressed slave does not acknowledge (ACKed) the transmitted data 0: The addressed slave acknowledged (ACKed) the transmitted data
4	GCNA	R	This field indicates that I2C in the master mode has sent a General Call, and no slave on the bus has acknowledged the General Call. 1: No slave informs responding with ACKed 0: A slave responds with ACKed
5	GCR	R	This field indicates that I2C in the master mode transmit a General Call, but the user programs the bytes after the General Call as bus read (DATA[9] is set to 1). 1: Connected to the bus read after GCALL 0: Failed to connect to the bus read after GCALL
6	HSAD	R	This field indicates that the master is in high-speed mode and the high-speed master code (error behavior) has been confirmed. 1: In HS mode, it is informed that the HS master cod has been received (ACKed) 0: In HS mode, it is not informed that the HS master code has been received (ACKed) - the scene does not exist
7	SNR	R	This field indicates that the master has sent a start (START) byte and the start byte has been acknowledged (error behavior). 1: ACK of start byte detected 0: ACK of start byte not detected
8	RNR10B	R	This field indicates that restart is disabled (IC_ RESTART_ EN bit (CTRL0[5])=0), and the master transmit a read instruction in 10-bit addressing mode. 1: When restart is disabled, the master attempts to read in 10-bit addressing mode 0: When restart is disabled, the master does not attempt to read in 10-bit addressing mode
9	MSTDIS	R	This field indicates that the user attempts to start the master operation when the master mode is disabled. 1: The user starts the master operation when MASTER is disabled 0: The user does not start the master operation when MASTER is disabled
10	ARBLOST	R	This field specifies that the master loses arbitration, and if TAS[14] is also set, the slave transmitter loses the arbitration. 1: The master or slave transmitter loses arbitration 0: The master or slave transmitter does not lose arbitration and there is no exception
11	LFTF	R	This field specifies that the slave receives a read instruction and there are some data in TXFIFO, so the slave transmit a TAIF interrupt to refresh the old data in TXFIFO. 1: Refresh the existing data in TX-FIFO after the slave machine obtains the read instruction 0: Refresh the existing data in TX-FIFO after the slave machine obtains the read instruction

Field	Name	R/W	Description
12	SAL	R	This field specifies that the slave loses arbitration 1: The slave transmitter loses arbitration 0: The slave transmitter does not lose arbitration
13	SRI	R	This field indicates that slave read done. 1: The slave read done 0: The slave does not read done
14	USRARB	R	This field indicates that the slave loses the bus when transmitting data to the remote master. At the same time, set TAS [12]. 1: The slave loses the bus 0: The slave does not lose the bus
15	FLUCNT	R	1: When the processor responds to the request of transmitting data to a remote master in save mode, the user writes 1 in the CMD (Bit 8) of the DATA register. 1: The slave attempts to transmit to remote master in read mode 0: The slave does not attempt to transmit to remote master in read mode
31:16	Reserved		

22.4.34 Generate slave data NACK register (SDNO)

Offset address: 0x84

Reset value: 0x00

Field	Name	R/W	Description
0	NACK	R/W	Generate NACK. This NACK generation occurs only when I2C is slave receiver. If this register is set to 1, it can generate a NACK only after receiving a data byte; therefore, data transmission is aborted and the received data will not be pushed to the receive buffer. When the register is set to 0, it generates NACK/ACK under normal conditions. 1 (ENABLED): Only NACK is generated when the slave receiver receives data 0 (DISABLED): Normal response after the slave receiver receives data
31:1	Reserved		

22.4.35 DMA control register (DMACTRL)

Offset address: 0x88

Reset value: 0x00

Field	Name	R/W	Description
0	RXEN	R/W	Receive DMA enable: This bit is used to enable/disable receive FIFO DMA channel. 1: Enable receive FIFO DMA channel 0: Disable receive FIFO DMA channel Note: The width of DMA data configured by the user can only be 16 bits and 32 bits.
1	TXEN	R/W	Transmit DMA enable: This bit is used to enable/disable the transmit FIFO DMA channel. 1: Enable FIFO DMA channel 0: Disable FIFO DMA channel Note: The width of DMA data configured by the user can only be 16 bits and 32 bits.
31:2	Reserved		

22.4.36 DMA transmitting data level register (DTDL)

Offset address: 0x8C

Reset value: 0x00

Field	Name	R/W	Description
2:0	DTDL	R/W	When the number of bytes in the transmit FIFO is less than or equal to the value indicated by this bit field, transmit the data level . This bit field controls the level of DMA request issued by the transmission logic. It equals to watermark level.
31:3	Reserved		

22.4.37 DMA receive data level register (DRDL)

Offset address: 0x90

Reset value: 0x00

Field	Name	R/W	Description
2:0	DRDL	R/W	Data level when the number of bytes in the receive FIFO is greater than the value indicated by this bit field; this field controls the level that the receive logic transmit DMA request, The watermark level=DRDL+1.
31:3	Reserved		

22.4.38 I2C SDA setup register (SDADLY)

Offset address: 0x94

Reset value: 0x64

Field	Name	R/W	Description
8:0	SDADLY	R/W	SDA setup It is suggested that if the required delay is 1000ns, for 10 MHz IC_CLK frequency, SDADLY should be programmed to be 11. The minimum input value of SDADLY is 2.
31:9	Reserved		

22.4.39 I2C ACK General Call register (GCA)

Offset address: 0x98

Reset value: 0x01

Field	Name	R/W	Description
0	GCA	R/W	ACK General Call When it is set to 1, I2C will respond with ACK when receiving General Call. Otherwise, I2C will respond with NACK. 1: General Call generates ACK 0: General Call generates NACK
31:1	Reserved		

22.4.40 I2C enable state register (STS2)

Offset address: 0x9C

Reset value: 0x00

Field	Name	R/W	Description
0	I2CEN	R	I2CEN Status。 This bit always reflects the value driven on the output port I2CEN. 1: Enable I2C 0: Disable I2C

Field	Name	R/W	Description
1	SDWB	R	Disable the slave when busy This bit indicates whether a potential or active slave operation is aborted because Bit 0 of the CTRL1 register is set from 1 to 0. When CPU writes 0 to CTRL1 register, set this bit, meanwhile: (a) Address byte of slave transmitter operation recevied by I2C from remote master. (b) Address and data bytes of slave operation from remote master. When it is read as 1, it is regarded that I2C executes a NACK by force in any link of I2C transmission, regardless of whether the I2C address matches the slave address in I2C (SLAADDR register), or whether the transmission is completed before CTRL1 is set to 0, but it has not been yet effective. Note: If the remote I2C master aborts the transmission before it has a chance to issue (NACK) on the I2C bus, and CTRL1[0] is set to 0, this bit is also set to 1. When it is read as 0, and the master is active or the I2C bus is idle, I2C is considered disabled. Note: When I2CEN (Bit 0) is read as 0, the CPU can safely read this bit. 1 (ACTIVE): Disable when the slave is active 0 (INACTIVE): Disable when the slave is idle
2	SRDL	R	Slave received data lost (Slave Received Data Lost) This bit indicates that the slave I2C transmits and receives the abort operation because Bit 0 of CTRL1 is set from 1 to 0. When read as 1, I2C is considered to be actively involved in an aborted I2C transmission (with a matching address), and the data transmitted by I2C has been input even if NACK has responded to the data byte. Note: If the remote I2C master aborts the transmission with stop (STOP) condition before I2C has a chance to NACK the transmission, and CTRL1[0] is set to 0, this bit is also set to 1. When it is read as 0, I2C is considered disabled and does not actively participate in the data phase of the slave receiver transmission. Note: When I2CEN (Bit 0) is read as 0, the CPU can safely read this bit. 1: Slave RX data lost 0: Slave RX data not lost
31:3	Reserved		

22.4.41 I2C SS, FS or FM+ spike suppression limit register (LSSSL)

Offset address: 0xA

Reset value: 0x05

Field	Name	R/W	Description
7:0	LSSSL	R/W	This register must be set before any I2C bus transaction to ensure stable operation. This register sets the duration (measured with IC_CLK cycle) of the longest spike filtered by the spike suppression logic in the SCL or SDA line. This register can only be written when I2C interface is disabled, corresponding to CTRL1[0] register set to 0. Writing at other time is invalid. The minimum effective value is 1; the value that hardware prevents writing is less than this value. If you attempt to tamper with this value, the result is set to 1.
31:8	Reserved		

22.4.42 I2C HS spike suppression limit register (HSSSL)

Offset address: 0xA4

Reset value: 0x01

Field	Name	R/W	Description
7:0	HSSSL	R/W	This register must be set during any I2C bus transaction to ensure stable operation. This register sets the duration (measured with IC_CLK cycle) of the longest spike filtered by the spike suppression logic in the SCL or SDA line; this register can write only when I2C interface is disabled, corresponding to CTRL1[0] register set to 0. Writing at other time is invalid. The minimum effective value is 1; the value that hardware prevents writing is less than this value. If you attempt to tamper with this value, the result is set to 1.
31:8	Reserved		

23 Serial Peripheral Interface/On-chip Audio Interface (SPI/I2S)

23.1 Full Name and Abbreviation Description of Terms

Table 95 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
Most Significant Bit	MSB
Least Significant Bit	LSB
Master Out Slave In	MOSI
Master In Slave Out	MISO
Serial Clock	SCK
Serial Data	SD
Master Clock	MCK
Word Select	WS
Pulse-code Modulation	PCM
Inter-IC Sound	I2S
Transmit	TX
Receive	RX
Busy	BSY

23.2 Introduction

SPI interface can be configured to support SPI protocol and I2S audio protocol. It works in SPI mode by default, and the functions can be switched in I2S mode through software.

Serial peripheral interface (SPI) provides data transmitting and receiving functions based on SPI protocol, which allows chips to communicate with external devices in half duplex, full duplex, synchronous and serial modes, and can work in master or slave mode.

The on-chip audio interface (I2S) supports four audio standards: Philips I2S standard, MSB alignment standard, LSB alignment standard and PCM standard. It can work in master/slave mode of half-duplex communication.

23.3 Main Characteristics

23.3.1 Main Characteristics of SPI

（1） Master and slave operation with 3-wire full duplex synchronous transmission and receiving

（2） Simplex synchronous transmission can be realized by two wires (the third bidirectional data line can be included/not included)

（3） Select 8-bitt or 16-bit transmission frame format

（4） Support multiple master device mode

（5） Support special transmission and receiving mark and can trigger interrupt

（6） Have SPI bus busy state flag

（7） Fast communication in master/slave mode, up to 18MHz

（8） Clock polarity and phase are programmable

（9） Data sequence is programmable; select MSB or LSB first

（10） Interrupt can be triggered by master mode fault, overrun and CRC error flag

（11） Have DMA transmit and receive buffers

（12） Calculation, transmission and verification can be conducted through hardware CRC

23.3.2 Main Characteristics of I2S

（1） Have master/slave mode of simplex communication (only transmit/receive)

（2） Four audio standards

- I2S Philips standard
- MSB alignment standard
- LSB alignment standard
- PCM standard

（3） 16/24/32-bit data length can be selected

（4） 16-bit or 32-bit channel length

（5） Clock polarity is programmable

（6） 16-bit data register is used for transmitting and receiving

（7） MSB is always the first in the data direction

（8） Transmitting and receiving supports DMA function

23.4 SPI Functional Description

23.4.1 Description of SPI Signal Line

Table 96 SPI Signal Line Description

Pin name	Description
SCK	Master device: SPI clock outputs Slave device: SPI clock inputs
MISO	Master device: Input the pin and receive data Slave device: Output the pin and send data Data direction: From slave device to master device
MOSI	Master device: Output the pin and send data Slave device: Input the pin and receive data Data direction: From master device to slave device
NSS	Software NSS mode: NSS pin can be used for other purposes. NSS mode of master device hardware: NSS output, single master mode. NSS closed output: Operation of multiple master environments is allowed. NSS mode of slave device hardware: NSS signal is set to low level as chip selection signal of slave.

23.4.2 Phase and Polarity of Clock Signal

The clock polarity and clock phase are CPOL and CPHA bits of SPI_CTRL1 register.

Clock polarity CPOL means the level signal of SCK signal line when SPI is in idle state.

- When CPOL=0, SCK signal line is in idle state and at low level
- When CPOL=1, SCK signal line is in idle state and at high level

Clock phase CPHA means the sampling moment of data

- When CPHA=0, the signal on MOSI or MISO data line will be sampled by the "odd edge" on SCK clock line.
- When CPHA=1, the signal on MOSI or MISO data line will be sampled by the "even edge" on SCK clock line.

SPI can be divided into four modes according to the states of clock phase CPHA and clock polarity CPOL.

Table 97 Four Modes of SPI

SPI mode	CPHA	CPOL	Sampling moment	Idle SCK clock
0	0	0	Odd edge	Low level
1	0	1	Odd edge	High level
2	1	0	Even edge	Low level

SPI mode	CPHA	CPOL	Sampling moment	Idle SCK clock
3	1	1	Even edge	High level

23.4.3 Data Frame Format

Set MSB or LSB to be first by configuring LSBSEL bit in SPI_CTRL1 register. Select to transmit/receive in 8/16-bit data frame format by configuring DFLSEL bit in SPI_CTRL1 register.

23.4.4 NSS Mode

Software NSS mode: Select to enable or disable this mode by configuring SSEN bit of SPI_CTRL1 register, and the internal NSS signal level is driven by ISSEL bit of SPI_CTRL1 register.

Hardware NSS mode:

- Turn on NSS output: When SPI is in master mode, enable SSOEN bit, NSS pin will be pulled to low level and SPI will automatically enter the slave mode.
- Turn off NSS output: Operation is allowed in multiple master environments.

23.4.5 SPI Mode

23.4.5.1 SPI master mode

In master mode, generate serial clock on SCK pin

Master mode configuration

- Configure MSMSEL=1 in SPI_CTRL1 register
- Select the polarity and phase by configuring CPOL and CPHA bits in SPI_CTRL1 register.
- Select 8/16-bit data frame format by configuring DFLSEL bit in SPI_CTRL1 register
- Select LSB or MSB first by configuring LSBSEL in SPI_CTRL1 register
- NSS configuration:
  - NSS pin works in input mode: in hardware mode, it is required to connect NSS pin to high level during the entire data frame transmission; in software mode, it is required to set SSEN bit and ISSEL bit in SPI_CTRL1 register
  - NSS works in output mode and it is required to configure SSOEN bit of SPI_CTRL2 register
- Enable SPI by configuring SPIEN bit in SPI_CTRL1 register

In master mode: MOSI pin is data output, which MISO is data input

23.4.5.2 SPI slave mode

In slave mode, SCK pin receives the serial clock transmitted from the master device

Configuration of slave mode

- Configure MSMSEL=0 in SPI_CTRL1 register
- Select the polarity and phase by configuring CPOL and CPHA bits in SPI_CTRL1 register.
- Select 8/16-bit data frame format by configuring DFLSEL bit in SPI_CTRL1 register
- Select LSB or MSB first by configuring LSBSEL in SPI_CTRL1 register
- NSS configuration:
  - In hardware mode: NSS pin must be at low level in the whole data frame transmission process
  - In software mode: Set SSEN bit in SPI_CTRL1 register and clear ISSEL bit
- Enable SPI by configuring SPIEN bit in SPI_CTRL1 register

In slave mode: MOSI pin is data input, which MISO is data output

23.4.5.3 Half-duplex communication of SPI

One clock line and one bidirectional data line

- Enable this mode by setting BMEN of SPI_CTRL1 register
- Control the data line to be input or output by setting BMOEN bit of SPI_CTRL1 register
- SCK pin is used as clock, MOSI pin is used in master device to transmit data, and MISO pin is used in slave device to transmit data

23.4.5.4 Simplex communication of SPI

One clock line and one unidirectional data line

In this mode, SPI module can only receive or only transmit

Send-only mode:

- Data are transmitted on send pin (MOSI in master mode, MISO in slave mode)
- Then the receive pin can be used as general I/O (MISO in master mode, MOSI in slave mode).

Receive-only mode:

- In master mode, enable SPI to initiate communication, clear SPEN pin of SPI_CTRL1 register, and it will stop receiving data immediately, not needing to read BSYFLG flag (always 1).
- Slave mode: When NSS is pulled to low level, SPI will receive all the time as long as SCK has clock pulse.

In receive-only mode, SPI output can be disabled by setting RXOMEN bit in SPI_CTRL1 register. At this time, release the transmit pin (MOSI in master mode, MISO in slave mode), which can be used for other functions.

23.4.6 Data Sending and Receiving Process in Different SPI Modes

Table 98 Run Mode of SPI

Mode	Configure	Data pin
Full duplex mode of master device	BMEN=0, RXOMEN=0	MOSI sends; MISO receives
Unidirectional receiving mode of master device	BMEN=0, RXOMEN=1	MOSI is not used; MISO receives
Bidirectional sending mode of master device	BMEN=1, BMOEN=1	MOSI sends; MISO is not used
Bidirectional receiving mode of master device	BMEN=1, BMOEN=0	MOSI is not used; MISO receives
Full duplex mode of slave device	BMEN=0, RXOMEN=0	MOSI receives, and MISO transmits
Unidirectional receiving mode of slave device	BMEN=0, RXOMEN=1	MOSI receives, while MISO is not used
Bidirectional sending mode of slave device	BMEN=1, BMOEN=1	MOSI is not used, and MISO transmits
Bidirectional receiving mode of slave device	BMEN=1, BMOEN=0	MOSI receives, while MISO is not used

Figure 95 Connection in Full Duplex Mode

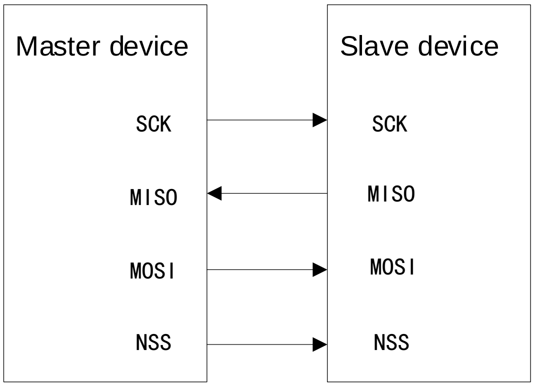


Figure 96 Connection in Simplex Mode (the master is used for receiving, while the slave is used for sending)

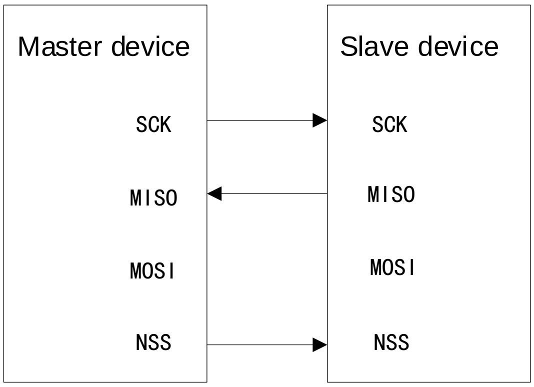


Figure 97 Connection in Simplex Mode (the master only sends, while the slave receives)

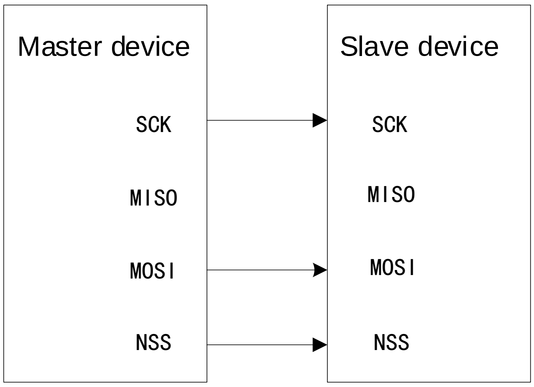
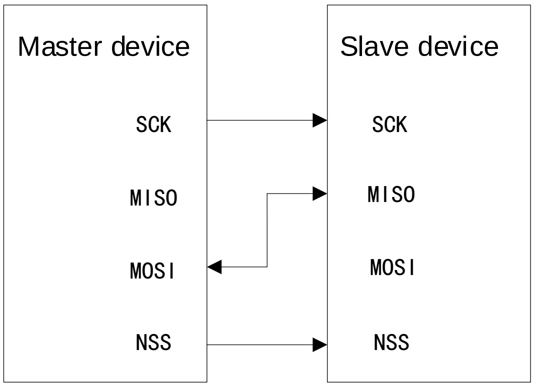


Figure 98 Bidirectional Line Connection



23.4.6.1 Transmitting and receiving of processed data

Data transmission

After the mode configuration is completed, the SPI module is enabled to remain idle.

Master mode: The software writes a data frame to the transmit buffer, and the transmission process starts

Slave mode: The SCK signal on the SCK pin starts to jump, while the NSS pin level is low, and the transmission process starts (before starting data transmission, make sure that the data has been written to the transmit buffer in advance).

When SPI is transmitting a data frame, it will load the data frame from the data buffer to the shift register, and then start to transmit data. After one bit of data frame is transmit, TXBEFLG is set to 1. If you need to continue to transmit data, the software needs to wait until TXBEFLG=1 writes data to the SPI_DATA register. (TXBEFLG flag is set to 1 by hardware and cleared by software).

Data receiving

BSYFLG flag is always set to 1 in the data transmission process.

At the last edge of the sampling clock, the received data is transferred from the shift register to the receive buffer; set the RXBNEFLG flag, and the software reads the data in data register (SPI_DATA) to obtain the content of the receive buffer; if RXBNEIEN bit of SPI_CTRL2 register is set, an interrupt will be generated, and after data is read, the BSYFLG flag will be automatically cleared.

23.4.6.2 Full duplex transmitting and receiving mode in master/slave device

Full duplex mode in master device

- After writing data to SPI_ DATA register (transmit buffer), data transmission starts.
- When SPI transmits the first bit of data, the data is transferred from the transmit buffer to the shift register and then transferred to the MOSI pin serially according to the sequence.
- The data received on MISO pin is serially transferred to SPI_DATA register (receive buffer) according to the sequence.

Transmitting and receiving are synchronous.

Full duplex mode under slave device

- When the slave device receives the clock signal and the first data bit appears on the MOSI pin, data transmission starts, and the subsequent data bits will be transferred to the shift register in turn.
- When SPI sends the first bit of data, the data is transferred from the transmit buffer to the shift register, and then transferred to the MISO pin serially according to the sequence.
- The software must ensure that the data to be transmit is written before the SPI master device starts to transmit data.

Sending and receiving are synchronous

Full duplex transmitting and receiving process under master/slave device

（1） Enable SPI module: Configure SPIEN=1 of SPI_CTRL1 register.

（2） Write the first data to be transmit to SPI_DATA register, and the TXBEFLG flag will be cleared.

（3） Wait until TXBEFLG flag bit is set to 1 (control by hardware), and write the second data bit to be transmit.

（4） Wait until RXBNEFLG flag bit is set to 1 (control by hardware), read the first received data in the SPI_DATA register, at the same time, clear the RXBNEFLG flag (cleared by software). Repeat the operation, and send and receive data at the same time.

（5） Wait until RXBNEFLG=1 and receive the last data.

（6） Wait until TXBEFLG=1 and disable SPI module after BSYFLG=0.

23.4.6.3 Bidirectional sending mode of master/slave device

Bidirectional transmission of master device

- Write data to SPI_ DATA register, and the transmission starts
- The data in the transmit buffer is transferred to the shift register in parallel, and then transferred to the MOSI pin serially according to the sequence.

Bidirectional transmission of slave device

- When the slave device receives the clock signal and the first data bit appears on the MISO pin, data transmission starts.
- At the same time, the data to be transmit by the transmit buffer is transferred to the shift register in parallel, and then transmit to the MISO pin in serial (before data transmission, make sure that the data has been written to the transmit buffer in advance).

Bidirectional transmission process of master/slave device

（1） Enable SPI module: Configure SPIEN=1 of SPI_CTRL1 register.

（2） Write the first data to be transmit to SPI_DATA register, and the TXBEFLG flag will be cleared.

（3） Wait until TXBEFLG=1, write the second data, repeat the operation and send the subsequent data

（4） After writing the last data, wait for TXBEFLG=1 and BSYFLG=0 and transmission is completed

23.4.6.4 One-way/Two-way receiving mode under master/slave device

（1） Enable SPI module: Configure SPIEN=1 of SPI_CTRL1 register.

（2） In the master device: Generate SCK clock immediately, and continuously receive data before SPI is disabled.

（3） Slave device: When SPI master device pulls down NSS and generates clock, receive data.

（4） Wait until the RXBNEFLG flag is set to 1, read data through SPI_DATA, and repeat the operation to receive data.

23.4.7 CRC Functions

SPI module contains two CRC computing units, which are used for data receiving and data transmission respectively.

CRC computing unit is used to define polynomials in SPI_CRCPOLY register.

Enable CRC computing by configuring CRCEN bit in SPI_CTRL1 register; at the same time, reset the CRC register (SPI_RXCRC and SPI_TXCRC).

To obtain the CRC value of transmission calculation, after the last data is written to the transmit buffer, it is required to set CRCNXT bit of SPI_CTRL1; indicate that the hardware sends the CRC value after the last data is transmit, and the CRCNXT bit will be cleared; at the same time, compare the values of CRC and SPI_RXCRC, and if they do not match, it is required to set CRCEFLG bit of SPI_STS register, and after ERRIEN bit of SPI_CTRL2 regiser is set, an interrupt will occur.

Note:

(1) If SPI is under slave device and CRC function is used, CRC computing will continue when NSS pin is at high level. For example, when the master device communicates with multiple slave devices alternately, the above situation will occur, so it is necessary to avoid faulty operation of CRC.

(2) In the process of a slave device from being unselected (NSS is at high level) to being selected (NSS is at low level 0), it is required to clear the CRC value at both ends of the master and slave devices to keep the next CRC computing results of the master and slave devices synchronized.

(3) When SPI is in slave mode, CRC computing can be enabled after the clock is stable.

(4) When the SPI clock frequency is too high, the CPU operation will affect the SPI bandwidth. It is recommended to use DMA mode to avoid the reduction of SPI speed.

(5) When the SPI clock frequency is too high, during the CRC transmission period, the CPU utilization frequency is reduced, and the function call is disabled in the CRC transmission process to avoid errors when receiving the last data and CRC.

(6) When NSS hardware mode is used in slave mode, NSS pin should be kept low during data transmission and CRC transmission period.

Sequence of clearing CRC values

（1） SPI Disabled (SPIEN=0)

（2） Clear CRCEN bit

（3） Set CRCEN bit to 1

（4） Enable SPI (SPIEN=1)

23.4.8 DMA Function

For high-speed data transmission, the request/response DMA mechanism in SPI improves the system efficiency and can transfer data to SPI transmit buffer promptly, and the receive buffer can read the data in time to prevent overflow.

When SPI only sends data, it is only needed to enable DMA transmission channel; when SPI only receives data, it is only needed to enable DMA receiving channel.

DMA function of SPI mode can be enabled by configuring TXDEN and RXDEN bits of SPI_CTRL2 register.

- When sending: When TXBEFLG flag bit is set to 1, issue the DMA request, DMA controller writes data to SPI_DATA register, and then the TXBEFLG flag bit will be cleared.
- When receiving: When setting RXBNEFLG flag bit to 1, issue the DMA request, DMA controller reads data from SPI_DATA register, and then RXBNEFLG flag bit is cleared.

By monitoring BSYFLG flag bit, confirm whether SPI communication is over after DMA has transferred all data to be transmit in sending mode, which can avoid damaging the transmission of last data.

DMA function with CRC

By the end of communication, if SPI enables both CRC operation and DMA function, sending and receiving of CRC bytes will be completed automatically.

At the end of data and CRC transmission, if CRCEFLG flag bit of SPI_STS register is set to 1, it indicates that an error occurred during transmission.

23.4.9 SPI Disable

After data transmission is over, end the communication by closing SPI module. In some configurations, if SPI is disabled before data transmission is completed, data transmission error may be caused. Different methods are required in different operation modes to disable SPI

Full duplex mode under master/slave device

（1） Wait until RXBNEFLG flag bit is set to 1, and receive the last data

（2） Wait until TXBEFLG flag bit is set to 1

（3） Wait for clearing BSYFLG flag bit

（4） Disable SPI (set SPIEN=0 of SPI_CTRL1 register)

One-way/Two-way receive-only mode under master/slave device

（1） Wait No. n-1 RXBNEFLG flag bit is set to 1

（2） Wait for one SPI clock cycle before SPI is disabled (set SPIEN=0 of SPI_CTRL1 register)

（3） Before entering the stop mode, wait until the last RXBNEFLG flag bit is set to 1

Receive-only/Two-way receiving mode in slave mode

SPI can be disabled at any time (set SPIEN=0 of SPI_CTRL1 register) and it will be disabled when the transmission is over. If you want to enter the stop mode, wait until BSYFLG flag bit is cleared.

23.4.10 SPI Interrupt

23.4.10.1 State flag bit

There are three flag bits for fully monitoring the state of SPI bus

Transmit buffer empty flag TXBEFLG

TXBEFLG=1 indicates that the transmit buffer bit is empty, and the next data to be transmit can be written. When the data is written to SPI_DATA register, clear the TXBEFLG flag bit.

Receive buffer non-empty flag RXBNEFLG

RXBNEFLG=1 indicates that the receive buffer contains valid data and the data can be read through SPI_DATA register; then clear the RXBNEFLG flag

Busy flag BSYFLG

BSYFLG flag is set and cleared by hardware, which can indicate the state of SPI communication layer. When BSYFLG=1, it indicates SPI is communicating, but in the two-line receiving mode under the master device, BSYFLG=0 during the period of receiving of data.

BSYFLG flag can be used to detect whether transmission is over to avoid damaging the last transmit data.

BSYFLG flag bit can be used to avoid conflict when writing data in multi-master mode.

BSYFLG flag will be cleared when the transmission ends (except for continuous communication in master mode), SPI is disabled and the master mode fails.

BSYFLG=0 between data item and data item when communication is discontinuous.

When communication is continuous:

- In master mode: BSYFLG=1 in the whole transmission process
- In save mode: BSYFLG is kept low within one SCK clock cycle between transmission of each data

Note: It is best to use TXBEFLG and RXBNEFLG flags to process the transmitting and receiving of each data item.

23.4.10.2 Error flag bit

Master mode error MEFLG

MEFLG is an error flag bit. The master mode error occurs when: in hardware NSS mode, the NSS pin of the master device is pulled down; in software NSS mode, ISSEL bit is cleared; MEFLG bit is set automatically.

Influence of master mode failure: MEFLG is set to 1, and SPI interrupt is generated when ERRIEN is set; SPIEN is cleared (output stops, SPI interface is disabled); MSMSEL is cleared and the device is forced into the slave mode.

Operation of clearing the MEFLG flag bit: When MEFLG flag bit is set to 1, it is required to read or write SPI_STS register, and then write to SPI_CTRL1 register.

When the MEFLG flag bit is 1, it is not allowed to set the SPIEN and MSMSEL bits

Overrun error OVRFLG

Overrun error: After the master device sends the data, the RXBNEFLG flag bit is still 1, which indicates that the overrun error occurred. Then OVRFLG bit is set to 1, and if the ERRIEN bit is also set, an interrupt will be generated.

After an overrun error occurs, the data in the receive buffer is not the data transmit by the master device, and then the read data in SPI_DATA register is the data not read before, while the data transmit later will not be read.

OVRFLG flag can be cleared by reading SPI_DATA register and SPI_STS register according to the sequence.

CRC error flag CRCEFLG

By setting CRCEN bit of SPI_CTRL1 register, start CRC computing, CRC error flag, and check whether the received data is valid.

When the value transmit by SPI_TXCRC register does not match the value in

SPI_RXCRC register, a CRC error will be generated, and CRCEFLG flag bit in SPI_STS register will be set to 1.

CRCEFLG can be cleared by writing 0 to CRCEFLG bit of SPI_STS register.

Table 99 SPI Interrupt Request

Interrupt flag	Interrupt event	Enable control bit	Clearing method
TXBEFLG	Transmit buffer empty flag	TXBEIEN	Write SPI_DATA register
RXBNEFLG	Receive buffer non-empty flag	RXBNEIEN	Read SPI_DATA register
MEFLG	Master mode failure event flag	ERRIEN	Read/Write SPI_STS register, and then write SPI_CTRL1 register
OVRFLG	Overrun error flag		Read SPI_DATA register, and then read SPI_STS register
CRCEFLG	CRC error flag		Write 0 to CRCEFLG bit

23.5 I2S Functional Description

Enable I2S function by setting I2SMOD bit of SPI_I2SCFG.

I2S and SPI share three pins:

- SD: Serial data, sending and receiving the data of 2-way time division multiplexing channel
- WS: Chip selection, switching the data of left and right channels
- CK: Serial clock; the clock signal is output in master mode, and is input in slave mode
- MCK: Master clock; in master mode, when MCOEN bit of SPI_I2SPSC register is set to 1, it can be used as the pin for outputting the extra clock signal.

23.5.1 I2S Audio Standard

I2S audio standard is selected by setting I2SSSEL bit and PFSSEL bit of SPI_I2SCFG register, and four audio standards can be selected: I2S Philips standard, MSB alignment standard, LSB alignment standard and PCM standard. Except PCM standard, other audio standards have two channels: left and right channels.

The data length and channel length can be configured by DATALEN and CHLEN bits in SPI_I2SCFG register. The channel length must be greater than or equal to the data length. There are four data formats to send data: 16-bit data packed into 16-bit frame, 16-bit data packed into 32-bit frame, 24-bit data packed into 32-bit frame, and 32-bit data packed into 32-bit frame.

When the 16-bit data is extended to 32 bits, the first 16 bits are valid data, and the last 16 bits are forced to be 0. No external intervention is needed in this process.

Since the data buffers used for sending and receiving are all 16 bits, SPI_DATA needs to read/write twice when 24-bit and 32-bit data are transmitted. If DMA is used, it needs to be transmit twice.

For all communication standards and data formats, the most significant bit of data is always transmitted first.

For time division multiplexing, the left channel is always transmitted first, and then the right channel is transmitted.

23.5.1.1 I2S Philips standard

In I2S Philips standard, the pin WS can indicate the data being transmitted comes from the left channel or the right channel.

In I2S Philips standard, both WS and SD change on the falling edge of CK clock signal.

The sender will change the data on the falling edge of the clock signal CK, while the receiver will change the data on the rising edge of the clock signal CK.

Figure 99 I2S Philips Protocol Waveform (16/32 bits)

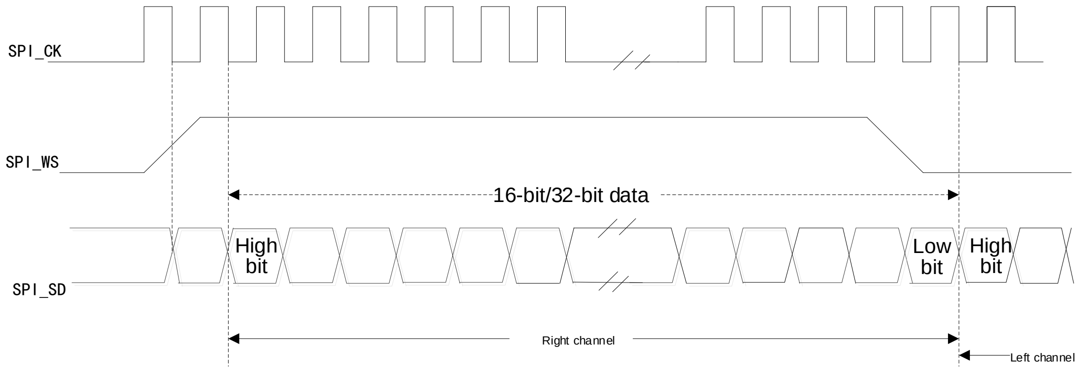
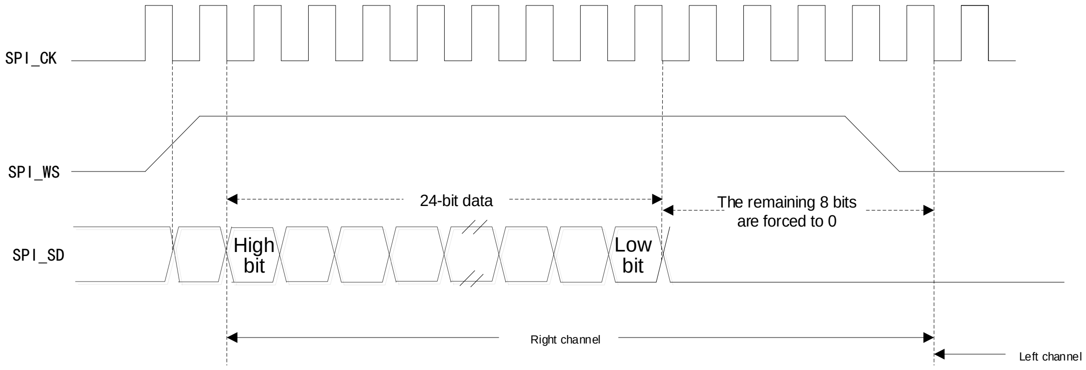


Figure 100 I2S Philips Protocol Waveform (24 bits)



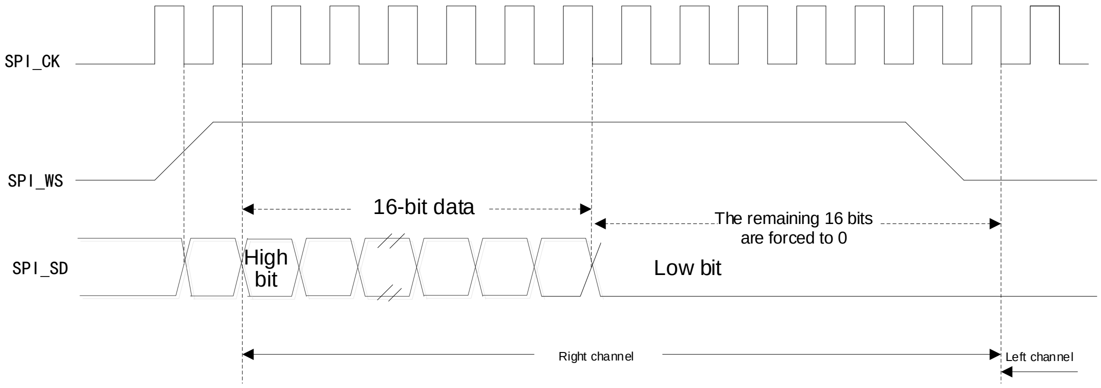
In I2S Philips standard, if you want to send/receive 24-bit and 32-bit data, the SPI_DATA register needs to read/write twice; for example:

- If you need to send 0x9FBB88 (24-bit data), write 0x9FBB to SPI_DATA register for the first time, and write 0x88XX to the register for the second time.
- If you need to receive 0x9FBB88 (24-bit data), read out 0x9FBB from SPI_DATA register for the first time and read out 0x8800 from the register for the second time.

In I2S configuration, when selecting the frame format of extending from 16-bit data to 32-bit data frame, it is required to access SPI_DATA register, and the remaining 16-bit data will be set to 0x0000 by hardware by forece; for example:

- The data to be received or transmit is 0x62D8, which becomes 0x62D80000 after it is expanded to 32 bits, and it is necessary to write 0x62D8 to SPI_DATA register or read out from SPI_DATA register.

Figure 101 I2S Philips Protocol Waveform (extending from 16 bits to 32 bits)



In the transmission process, the MSB should be written to the register SPI_DATA, and when TXBEFLG flag bit is set to 1, new data can be written; if there is corresponding interrupt, an interrupt can be generated.

In the receiving process, every time the MSB is received, the RXBNEFLG flag bit will be set to 1; if there is corresponding interrupt, an interrupt can be generated.

23.5.1.2 MSB alignment standard

In MSB standard, WS signal and the first data bit are generated at the same time

In the transmission process, the data is changed on the falling edge of the clock signal; in the receiving process, the data is read on the rising edge of the clock signal.

Figure 102 MSB Alignment Standard Waveform (16/32-bit data)

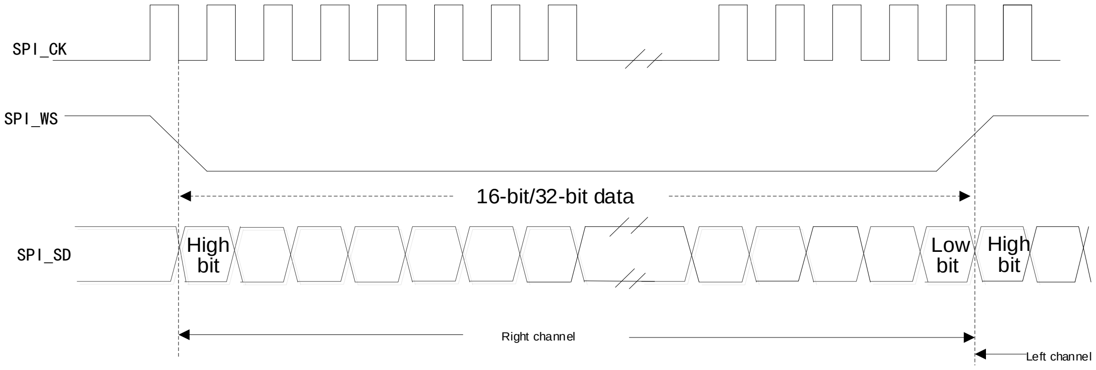


Figure 103 MSB Alignment Standard Waveform (24-bit data)

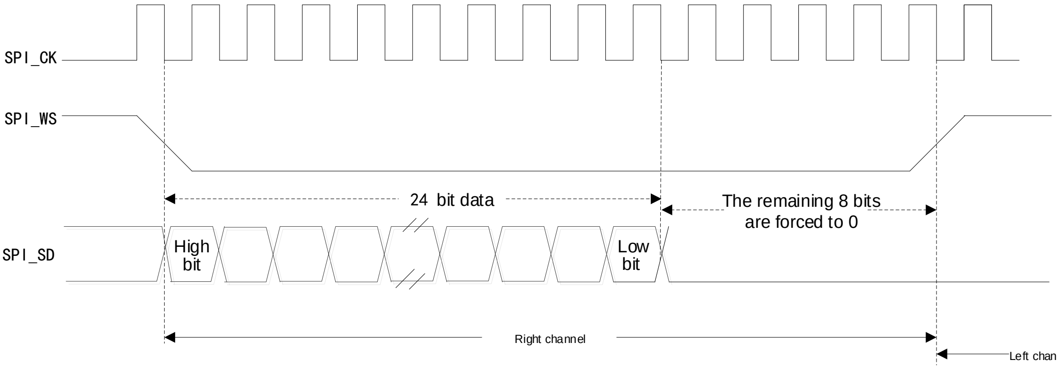
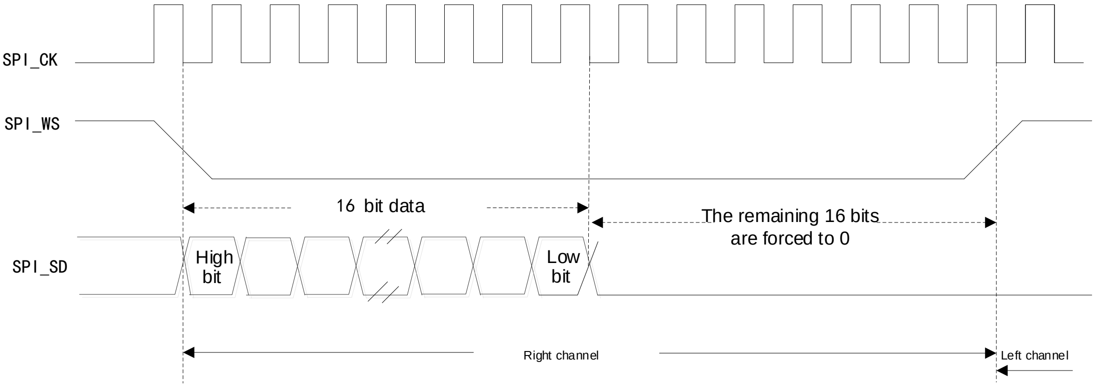


Figure 104 MSB Alignment Standard Waveform (extending from 16 bits to 32 bits)



23.5.1.3 LSB alignment standard

In the transmission process of LSB alignment standard, the data is changed on the falling edge of the clock signal; in the receiving process, the data is read on the rising edge of the clock signal. When the channel length is the same as the data length, the LSB alignment standard is the same as the MSB alignment standard. If the channel length is larger than the data length, the valid data of the LSB alignment standard is aligned with the lowest bit.

Figure 105 LSB Alignment Standard Waveform (16/32-bit data)

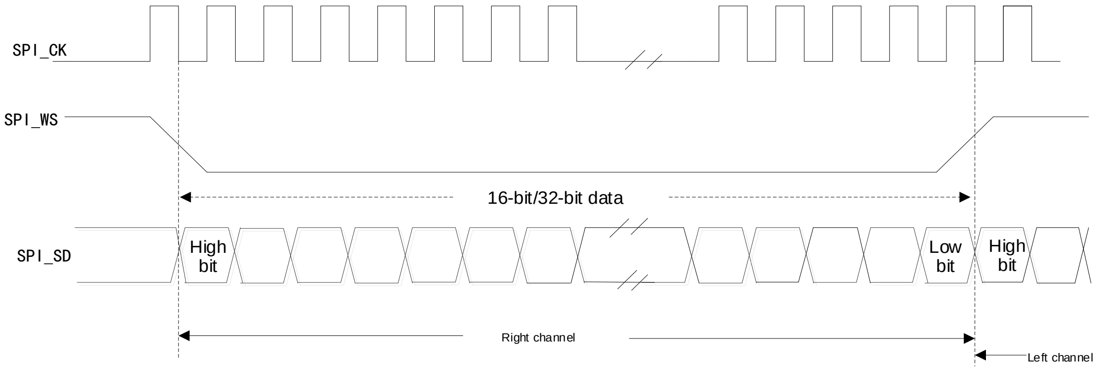
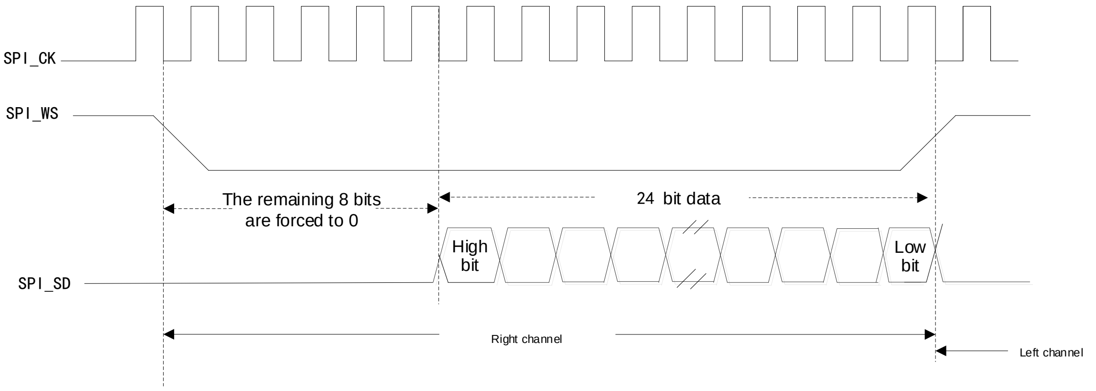


Figure 106 LSB Alignment Standard Waveform (24-bit data)



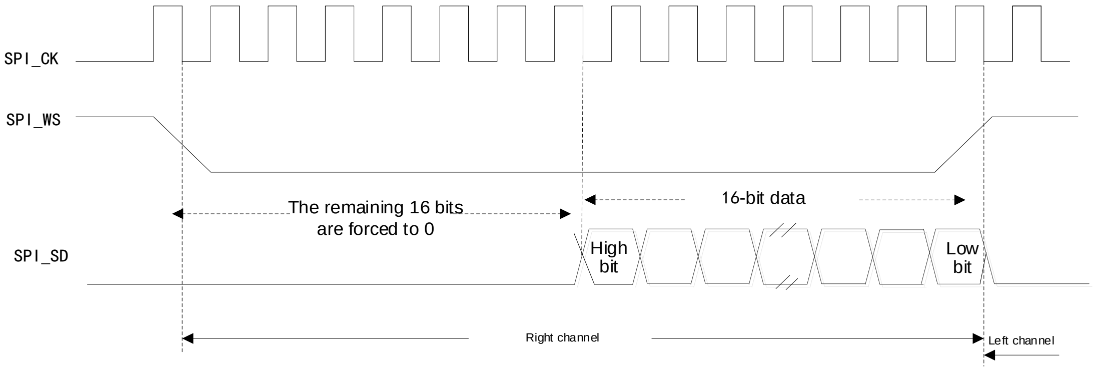
In the transmission process, if you want to send/receive 24-bit data, it is required to read/write the SPI_DATA register twice; for example:

- When you need to send 0x56EA98, write 0xXX56 to SPI_DATA register for the first time, and write 0xEA98 to SPI_DATA for the second time.
- When you need to receive 0x56EA98, read out 0x0056 from SPI_DATA registr for the first time, ad read out 0xEA98 from SPI_DATA register for the second time.

In I2S configuration, when selecting the frame format of extending from 16-bit data to 32-bit data frame, it is required to access SPI_DATA register, and the high 16-bit data will be set to 0x0000 by hardware by forece; for example:

- The data to be received or transmit is 0x98A5, which becomes 0x000098A5 after it is expanded to 32 bits, and it is necessary to write 0x98A5 to SPI_DATA register or read out from SPI_DATA register.

Figure 107 Under LSB Alignment Standard (extending from 16 bits to 32 bits)

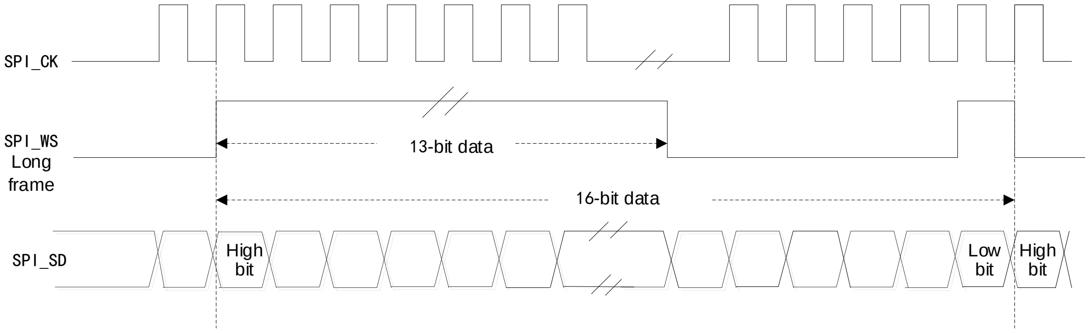


23.5.1.4 PCM standard

There is no sound channel selection in PCM standard. Short frame and long frame of PCM standard are selected by configuring PFSSEL bit in SPI_I2SCFG register.

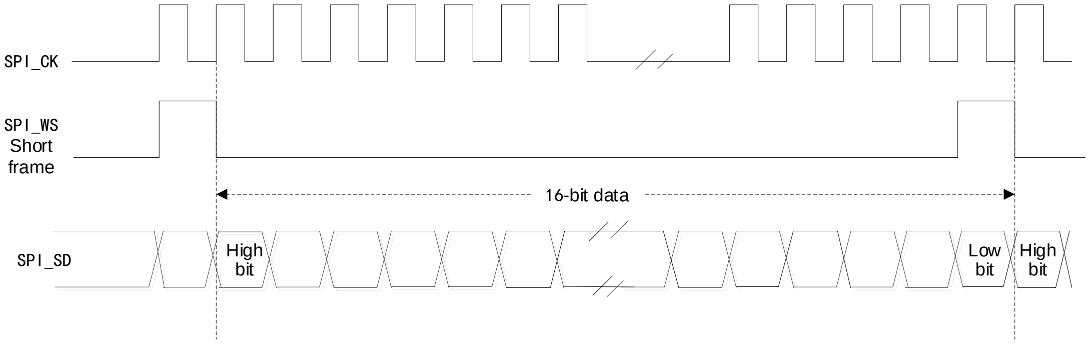
In the master mode, the valid time of synchronous WS signal of the long frame structure is 13 bits.

Figure 108 PCM Standard Waveform



In the master mode, the length of the synchronous WS signal of the short frame structure is 1 bit.

Figure 109 PCM Standard Waveform



23.5.2 I2S Clock

The clock source of I2SxCLK is system clock (HSICLK, HSECLK or PLL of AHB clock)

The bit rate of I2S determines the data stream on I2S data line and the clock signal frequency of I2S.

- I2S bit rate = the number of bits per channel × the number of sound channels × audio sampling frequency
- There are two channels of 16 bit audio signal: I2S bit rate=16×2×Fs

The relationship between audio sampling frequency (Fs) and I2S bit rate (I2S) is defined by the following formula:

Table 100 Audio Sampling Frequency (Fs) Formula

MCOEN	CHLEN	Audio sampling frequency (Fs)
1	0	I2SxCLK/[（16*2）*（（2*I2SPSC）+ODDPSC）*8]
1	1	I2SxCLK/[（32*2）*（（2*I2SPSC）+ODDPSC）*4]
0	0	I2SxCLK/[（16*2）*（（2*I2SPSC）+ODDPSC）]
0	1	I2SxCLK/[（32*2）*（（2*I2SPSC）+ODDPSC）]

23.5.3 I2S Mode

Table 101 I2S Run Mode

Run mode	SD	WS	CK	MCK
Master transmitting	Output	Output	Output	Output/Not used
Master receiving	Input	Output	Output	Output/Not used
Slave transmitting	Output	Input	Input	Output/Not used
Slave receiving	Input	Input	Input	Output/Not used

23.5.3.1 I2S master mode configuration process

（1） Configure I2SPSC bit and ODDPSC bit of SPI_I2SPSC register to define the baud rate of serial clock and the actual frequency division factor corresponding to the audio sampling frequency.

（2） Configure CPOL bit of SPI_I2SCFG register to define the clock polarity of SPI in idle state.

（3） Configure I2SMOD bit of SPI_I2SCFG register to activate I2S function and configure I2SMOD and PFSSEL bits of SPI_I2SCFG register to select I2S standard; configure DATALEN bit of SPI_I2SCFG register to select the data bits of the sound channel, and configure I2SMOD bit to select I2S master mode as transmitting terminal/receiving terminal.

（4） Configure SPI_CTRL2 register to select to enable the interrupt and DMA function or not (select required or not).

（5） Configure WS pin and CK pin to output mode; when MCOEN bit of SPI_I2SPSC is set to 1, the MCK pin should also be configured to output mode.

（6） Set the running mode of I2S by configuring the I2SMOD bit of SPI_I2SCFG.

（7） Set I2SEN bit of SPI_I2SCFG register to 1.

23.5.3.2 I2S master mode transmission process

When the data is written to the transmit buffer, the transmission will start, and the data will be transferred from the transmit buffer to the shift register, the TXBEFLG flag position is set to 1, and the SCHDIR flag bit indicates the corresponding sound channel of the currently transmit data. And the value of SCHDIR flag bit will be updated when TXBEFLG flag bit is 1.

When sending the first bit of data, 16-bit data will be transferred to the 16-bit shift register in parallel, and then transmit from the pin MISO/SD in serial. The next data needs to be written to SPI_DATA register when TXBEFLG flag bit is 1. If TXBEIEN bit of SPI_CTRL2 is 1, an interrupt will be generated.

Before the completion of the current data transmission, write the next data to be transmit to ensure continuous transmission of audio data.

When I2S is disabled, I2SEN can be cleared only when the flag bit TXBEFLG is 1 and BSYFLG is 0.

23.5.3.3 I2S master mode receiving process

RXBNEFLG flag is used to control the receiving sequence. RXBNEFLG flag indicates whether the receive buffer is empty; when the receive buffer is full, the RXBNEFLG flag bit will be set to 1. If RXBNEIEN bit ofSPI_CTRL2 is configured, an interrupt will occur and after the user reads out the data from SPI_DATA register, the RXBNEFLG flag bit will be cleared. Make sure to receive new data after reading operation; otherwise, overrun will occur and the OVRFLG flag bit will be set to 1.

The value of SCHDIR should be updated immediately after receiving data, and it depends on the WS signal generated by I2S.

Regardless of the data type and the channel length, the audio data is always received in the form of 16 bits. According to the configured data and the length of the channel, the data needs to be transmit to the receive buffer once or twice.

Turn off the I2S function, and for different audio protocols, the data length and channel length operation steps are as follows:

The data length is 16 bits, and 32-bit channel length (DATALEN=00, CHLEN=1, I2SSSEL=10) in LSB alignment mode

- Wait until the penultimate RXBNEFLG is set to 1
- Wail for 17 I2S clock cycles (software delay)
- I2SEN flag bit is cleared

The data length is 16 bits, and 32-bit channel length (DATALEN=00, CHLEN=1, I2SSSEL=10) in MSB alignment mode

- Wait until the last RXBNEFLG is set to 1
- Wail for one I2S clock cycle (software delay)
- I2SEN flag bit is cleared

All the other situations

- Wait until the penultimate RXBNEFLG is set to 1
- Wail for one I2S clock cycle (software delay)
- I2SEN flag bit is cleared

BSYFLG flag clock is low during data transmission

23.5.3.4 I2S slave mode configuration process

The configuration method of slave mode is basically the same as that of master mode. In slave mode, the clock signal and WS signal are provided by external I2S device instead of I2S.

（1） Configure I2SMOD bit of SPI_I2SCFG register to activate I2S function.

（2） Configure I2SSSEL bit of SPI_I2SCFG register to select the I2S standard; configure DATALEN[1:0] bit of SPI_I2SCFG register to select the bits of data; configure CHLEN bit of SPI_I2SCFG register to select the data bits per channel; configure I2SMOD bit of SPI_I2SCFG register to select I2S slave mode as transmitting terminal/receiving terminal.

（3） Configure SPI_CTRL2 register to select to enable the interrupt and DMA function or not (select required or not).

（4） Set I2SEN bit of SPI_I2SCFG register to 1.

23.5.3.5 I2S slave mode transmission process

Enable the slave device, write the data to the I2S data register, the external master device will start to communicate, and the external master device will send the clock signal, and when the data transmission starts, the sending process will begin.

When the first bit data is transmit, the 16-bit data will be transferred to the 16-bit shift register in parallel, and then transmit from the pin MOSI/SD in series. When the data is transferred from the data register to the shift register, the TXBEFLG flag bit is set to 1; at this time if TXBEIEN bit of SPI_CTRL2 register is set, an interrupt will be generated. In order to ensure the continuity of data transmission, the next data should be written to SPI_DATA register before the data transmission is completed; otherwise, "underrun" will occur, and the UDRFLG flag bit will be set to 1.

SCHDIR bit of SPI_STS register indicates the channel corresponding to the transmit data. In the slave mode, the SCHDIR bit is determined by the WS signal of the external master device.

In MSB and LSB alignment mode of I2S, the first data written to the data register corresponds to the data of the left channel.

Disable I2S, and after the TXBEFLG flag bit is set to 1, BSYFLG flag bit can be cleared.

23.5.3.6 I2S slave mode receiving process

RXBNEFLG flag is used to control the receiving sequence. The RXBNEFLG flag indicates whether the receive buffer is empty; after the receive buffer is full, the RXBNEFLG flag bit will be set to 1; if RXBNEIEN bit of SPI_CTRL2 register is set, an interrupt will occur, and after the data are read out from SPI_DATA register, RXBNEFLG flag bit will be cleared; make sure to receive new data after read operation; otherwise, "overrun" will occur, and the OVRFLG flag bit will be set to 1.

The value of SCHDIR should be updated immediately after receiving data, and it depends on the WS signal generated by I2S.

Regardless of the data type and the channel length, the audio data is always received in the form of 16 bits. According to the configured data and the length of the channel, the data needs to be transmit to the receive buffer once or twice.

Disable I2S, and when receiving the last RXBNEFLG set to 1, I2SEN flag bit will be cleared.

23.5.4 I2S Interrupt

23.5.4.1 State flag bit

There are three state flag bits in I2S to monitor the state of I2S bus.

Transmit buffer empty flag bit TXBEFLG

When the TXBEFLG flag bit is 1, it indicates that the transmit buffer is empty, and the data to be transmit can be written to the transmit buffer; after data is written, the TXBEFLG flag bit will be cleared. (When I2S is disabled, the TXBEFLG flag bit is 0).

Receive buffer non-empty flag bit RXBNEFLG

When the RXBNEFLG flag bit is 1, it indicates that the receive buffer has data to be received; after read operation is performed on the SPI_DATA register, RXBNEFLG flag bit will be cleared.

Busy flag bit BSYFLG

When the BSYFLG flag bit is 1, it indicates that I2S is in communication state (set and cleared by hardware), but in the master receiving mode, the BSYFLG flag bit is always 0 during the receiving period.

When I2S is disabled and data transmission is over, the BSYFLG flag bit will be cleared.

During continuous communication

- In the master sending mode, the BSYFLG flag bit is always high during the transmission period.
- In the slave mode, during transmission of each data item, the BSYFLG flag bit is set to 0 within one I2S clock cycle.

Channel flag bit SCHDIR

In the sending mode, the SCHDIR flag bit indicates the data transmit on the SD pin is in the left channel or the right channel. This flag bit is refreshed when TXBEFLG=1.

In the sending process of slave mode, if there is an underrun error, the value of SCHDIR flag bit will be invalid. If needing to restart the communication, the I2S function should be turned off and then turned on.

In the receiving mode, the SCHDIR flag bit indicates the received data is from the left channel or the right channel. This flag bit is refreshed when SPI_DATA register receives data.

If there is an underrun error in the receiving mode, the value of SCHDIR flag bit will be invalid. If needing to restart the communication, the I2S function should be turned off and then turned on.

As there is no channel selection in PCM standard, the SCHDIR flg bit is meaningless.

When OVRFLG and UDRFLG flag bits of SPI_STS register is 1 and ERRIEN=1 for SPI_CTRL2, interrupt will be generated. The interrupt flag can be cleared by reading the value of SPI_STS register.

23.5.4.2 Error flag bit

I2S includes two error flag bits

Underrun flag bit UDRFLG

In the sending mode, if new data to be transmit is written to SPI_DATA register before the data is transmit, UDRFLG flag bit will be set to 1; at this time if ERRIEN bit of SPI_CTRL2 register is set to 1, an interrupt will be generated.

This flag bit will take effect only after I2SMOD bit of SPI_I2SCFG is set to 1. Clear the UDRFLG flag bit by reading SPI_STS register.

Overrun flag bit OVRFLG

In the receiving mode, if a new data is received before the data is read, OVRFLG flag bit will be set to 1. At this time if ERRIEN bit of SPI_CTRL2 register is set to 1, an interrupt will be generated, indicating the occurrence of the error.

Read SPI_DATA register to return the last correctly received data, and all the other newly received data will be lost. OVRFLG flag can be cleared by first reading SPI_STS register and then reading SPI_DATA register.

Table 102 I2C Interrupt Request

Interrupt flag	Interrupt event	Enable control bit	Clearing method
TXBEFLG	Transmit buffer empty flag	TXBEIEN	Write SPI_DATA register
RXBNEFLG	Receive buffer non-empty flag	RXBNEIEN	Read SPI_DATA register
OVRFLG	Underrun flag bit	ERRIEN	Read SPI_STS register
UDRFLG	Overrun flag bit		Read SPI_STS register Read SPI_DATA register again

23.5.4.3 DMA function

In I2S mode, the work mode of DMA is the same as that of SPI, except that it does not support CRC function.

23.6 Register Address Mapping

Table 103 SPI and I2S Register Mapping

Register name	Description	Offset address
SPI_CTRL1	SPI control register 1	0x00
SPI_CTRL2	SPI control register 2	0x04
SPI_STS	SPI state register	0x08
SPI_DATA	SPI data register	0x0C
SPI_CRCPOLY	SPI CRC polynomial register	0x10
SPI_RXCRC	SPI receive CRC register	0x14
SPI_TXCRC	SPI transmit CRC register	0x18
SPI_I2S_CFG	SPI I2S configuration register	0x1C
SPI_I2SPSC	SPI I2S prescaler register	0x20

23.7 Register Functional Description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

23.7.1 SPI control register 1 (SPI_CTRL1) (not used in I2S mode)

Offset address: 0x00

Reset value: 0x0000

Field	Name	R/W	Description
0	CPHA	R/W	Clock Phase Configure This bit indicates on the edge of which clock to start sampling 0: On the edge of No. 1 clock 1: On the edge of No. 2 clock Note: This bit cannot be modified during communication.
1	CPOL	R/W	Clock Polarity Configure Level state maintained by SCK when SPI is in idle state. 0: Low level 1: High level Note: This bit cannot be modified during communication
2	MSMCFG	R/W	Master/Salve Mode Configure 0: Configure as slave mode 1: Configure as master mode Note: This bit cannot be modified during communication
5:3	BRSEL	R/W	Baud Rate Divider Factor Select 000：DIV=2 001：DIV=4 010：DIV=8 011：DIV=16 100：DIV=32 101：DIV=64 110：DIV=128 111：DIV=256 Baud rate=FPCLK/DIV Note: This bit cannot be modified during communication
6	SPIEN	R/W	SPI Device Enable 0: Disable 1: Enable Note: When SPI device is disabled, please operate according to the process of closing SPI.
7	LSBSEL	R/W	LSB First Transfer Select 0: First send the most significant bit (MSB) 1: First send the least significant bit (LSB)
8	ISSEL	R/W	Internal Slave Device Select When CTRL1_SSEN=1 (software NSS mode), select internal NSS level by configuring the bit 0: Internal NSS is low 1: Internal NSS is high
9	SSEN	R/W	Software Slave Device Enable 0: Software NSS mode is disabled, and the internal NSS level is determined by external NSS pin 1: Software NSS mode is enabled, and the internal NSS level is determined by external NSS pin

Field	Name	R/W	Description
10	RXOMEN	R/W	Receive Only Mode Enable 0: Transmit and receive at the same time 1: Receive-only mode RXOMEN bit and BMEN bit together determine the transmission direction in the two-line and two-way mode. In the configuration of multiple slave devices, in order to avoid data transmission conflict, it is necessary to set RXOMEN bit to 1 on the slave devices that are not accessed.
11	DFLSEL	R/W	Data Frame Length Format Select 0: 8-bit data frame format 1: 16-bit data frame format Only when SPIEN=0, can this bit be written to change the data frame length.
12	CRCNXT	R/W	CRC Transfer Next Enable 0: The next transmitted data is from transmit buffer 1: The next transmitted data is from CRC register Note: After the last data is written to SPI_DATA register, set CRCNXT bit immediately.
13	CRCEN	R/W	CRC Calculate Enable 0: Disable 1: Enable CRC check function only applies to full duplex mode; only when SPIEN=0, can this bit be changed.
14	BMOEN	R/W	Bidirectional Mode Output Enable 0: Disable, namely, receive-only mode 1: Enable, namely, transmit-only mode When BMEN=1, namely, in single-line/double-line mode, this bit decides the transmission direction of transmission line.
15	BMEN	R/W	Bidirectional Mode Enable 0: Double-line unidirectional mode 1: Single-line bidirectional mode Single-line two-way transmission means: the transmission between MOSI pin of data master and MISO pin of slave

23.7.2 SPI control register 2 (SPI_CTRL2)

Offset address: 0x04

Reset value: 0x0000

Field	Name	R/W	Description
0	RXDEN	R/W	Receive Buffer DMA Enable When RXDEN=1, once RXBNEFLG flag is set, DMA request will be issued. 0: Disable 1: Enable

Field	Name	R/W	Description
1	TXDEN	R/W	Transmit Buffer DMA Enable When this bit is set, once TXBEFLG flag is set, DMA request will be issued. 0: Disable 1: Enable
2	SSOEN	R/W	SS Output Enable SS output in master mode 0: SS output is disabled, and it can work in multi-master mode. 1: SS output is enabled, and it cannot work in multi-master mode. Note: Not used in I2S mode.
4:3	Reserved		
5	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable When an error occurs, ERRIEN bit controls whether to generate the interrupt.
6	RXBNEIEN	R/W	Receive Buffer Not Empty Interrupt Enable 0: Disable 1: Enable When RXBNEFLG flag bit is set to 1, an interrupt request will be generated
7	TXBEIEN	R/W	Transmit Buffer Empty Interrupt Enable 0: Disable 1: Enable When TXBEFLG fag bit is set to 1, an interrupt request will be generated
15:8	Reserved		

23.7.3 SPI state register (SPI_STS)

Offset address: 0x08

Reset value: 0x0002

Field	Name	R/W	Description
0	RXBNEFLG	R	Receive Buffer Not Empty Flag 0: Empty 1: Not empty
1	TXBEFLG	R	Transmit Buffer Empty Flag 0: Not empty 1: Empty
2	SCHDIR	R	Sound Channel Direction Flag 0: Indicate that the left channel is transmitting or receiving the required data 1: Indicate that the right channel is transmitting or receiving the required data Note: Not used in SPI mode, without left and right channels in PCM mode.

Field	Name	R/W	Description
3	UDRFLG	R	Underrun Occur Flag 0: Not occur 1: Occurred This flag bit is set by hardware, and it can be cleared by writing 0 to this bit by software. Not used in SPI mode
4	CRCEFLG	RC_W0	CRC Error Occur Flag This bit indicates whether the received CRC value matches the value of RXCRC register 0: Match 1: Not match This bit is set by hardware, can be cleared by writing 0 to this bit by software, and is not used in I2S mode.
5	MEFLG	R	Mode Error Occur Flag 0: Not occur 1: Occurred This bit is set by hardware, can be cleared by writing 0 to this bit by software, and is not used in I2S mode.
6	OVRFLG	R	Overrun Occur Flag 0: Not occur 1: Occurred This bit is set by hardware, and it can be cleared by writing 0 to this bit by software.
7	BSYFLG	R	SPI Busy Flag 0: SPI is idle 1: SPI is communicating It is set or cleared by hardware.
15:8	Reserved		

23.7.4 SPI data register (SPI_DATA)

Offset address: 0x0C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	DATA	R/W	Transmit Receive Data register When writing this register, the data will be written to the transmit buffer; when reading this register, the data in receive buffer will be read. The size of the buffer is consistent with the length of the data frame, that is, for 8-bit data, only DATA[7:0] is used when sending and receiving data, and DATA[15:8] is invalid; for 16-bit data, DATA[15:0] will be used when sending and receiving data.

23.7.5 SPI CRC polynomial register (SPI_CRCPOLY) (not used in I2S mode)

Offset address: 0x10

Reset value: 0x0007

Field	Name	R/W	Description
15:0	CRCPOLY	R/W	CRC Polynomial Value Setup This register contains CRC polynomial of CRC computing, which can be modified, and the reset value is 0x0007.

23.7.6 SPI receive CRC register (SPI_RXCRC) (not used in I2S mode)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
15:0	RXCRC	R	Receive Data CRC Value The CRC data of receive data calculated by hardware are stored in this register; the bits and the length of data frames are consistent, that is, if the received data are 8 bits, the CRC computing is made based on CRC8; if the received data are 16 bits, the CRC computing is made based on CRC16. When CRCEN is set, the hardware clears the register. Note: When BSYFLG bit is set to 1, the value of reading RXCRC register may be wrong.

23.7.7 SPI transmit CRC register (SPI_TXCRC)

Offset address: 0x18

Reset value: 0x0000

Field	Name	R/W	Description
15:0	TXCRC	R	Transmit Data CRC Value The CRC data of transmitted data calculated by hardware are stored in this register; the bits and the length of data frames are consistent, that is, if the transmitted data are 8 bits, the CRC computing is based on CRC8; if the transmitted data are 16 bits, the CRC computing is based on CRC16. When CRCEN is set, the hardware clears the register. Note: When BSYFLG bit is set to 1, the value of reading RXCRC register may be wrong.

23.7.8 SPI_I2S configuration register (SPI_I2SCFG)

Offset address: 0x1C

Reset value: 0x0000

Field	Name	R/W	Description
0	CHLEN	R/W	Channel Length Configure The channel length refers to the data bits per audio channel 0: 16-bit width 1: 32-bit width The vocal tract length can be configured successfully only when the vocal tract length is greater than the data length; otherwise, the hardware will automatically adjust the vocal tract length; this bit can only be configured when I2SEN=0, and is not used in SPI mode.

Field	Name	R/W	Description
2:1	DATALEN	R/W	Configure the Length of the sData to Be Transferred 00: 16-bit data length 01: 24-bit data length 10: 32-bit data length 11: Not allowed This bit can only be configured when I2SEN=0, and is not used in SPI mode.
3	CPOL	R/W	Idle State Clock Polarity Configure 0: Low level 1: High level This bit can only be configured when I2SEN=0, and is not used in SPI mode.
5:4	I2SSSEL	R/W	I2S Standard Selection 00: I2S Philips standard 01: High-byte alignment standard (left alignment) 10: Low-byte alignment standard (right alignment) 11: PCM standard This bit can only be configured when I2SEN=0, and is not used in SPI mode.
6	Reserved		
7	PFSSEL	R/W	PCM Frame Synchronization Mode Select 0: Synchronization of short frames 1: Synchronization of long frames Apply only to PCM standard (I2SSSEL=11); this bit can only be configured when I2SEN=0, and is not used in SPI mode.
9:8	I2SMOD	R/W	I2S Master/Slave Transmit/Receive Mode Configure 00: Slave device transmits 01: Slave device receives 10: Master device transmits 11: Master device receives This bit can only be configured when I2SEN=0, and is not used in SPI mode.
10	I2SEN	R/W	I2S Enable 0: I2S is disabled 1: I2S is enabled Note: It is not used in SPI mode.
11	MODESEL	R/W	SPI/I2S Mode Select 0: Select SPI mode 1: Select I2S mode Note: This bit can be set only when SPI or I2S is disabled.
15:12	Reserved		

23.7.9 SPI_I2S prescaler register (SPI_I2SPSC) (not used in SPI mode)

Offset address: 0x20

Reset value: 0x0002

Field	Name	R/W	Description
7:0	I2SPSC	R/W	I2S Linear Prescaler Factor Configure I2SPSC cannot be set to 0 and 1; this bit can be configured only when I2SEN=0, and it is not used in SPI mode.
8	ODDPSC	R/W	Configure the prescaler factor to be odd 0: Actual division factor=I2SPSC*2 1: Actual division factor=(I2SPSC*2)+1 This bit can only be configured when I2SEN=0, and is not used in SPI mode.
9	MCOEN	R/W	Master Device Clock Output Enable 0: Disable 1: Enable This bit can only be configured when I2SEN=0, and is not used in SPI mode.
15:10	Reserved		

24 Controller Area Network (CAN)

24.1 Full Name and Abbreviation Description of Terms

Table 104 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
First Input First Output	FIFO
Request	REQ

24.2 Introduction

CAN is abbreviation of Controller Area Network, and is serial communication protocol of ISO international standardization and supports CAN Protocol 2.0A and 2.0B. In CAN protocol, the sender sends the message to all receivers in the form of broadcast. When the node receives the message, it will go through the filter group and decide whether the message is needed according to the identifier. This design saves the CPU overhead.

24.3 Main Characteristics

（1） Support CAN protocol 2.0A and 2.0B

（2） The maximum baud rate of communication is 1Mbit/s

（3） Transmission function

- There are three transmitting mailboxes
- The priority of transmitting message can be configured
- Record the transmission time

（4） Receiving function

- Have two receive FIFO with three depth levels
- Have 28 filter groups.
- Record the receiving time

（5） Memory

- CAN1 and USBD1 share 512-Byte SRAM
- CAN2 and USBD2 share 512-Byte SRAM
- CAN1 and USBD2 are used at the same time
- CAN2 and USBD1 are used at the same time
- CAN1 and CAN2 are used at the same time

24.4 Functional Description

24.4.1 Characteristics of CAN Physical Layer

There can be multiple communication nodes on the CAN bus, each node consists of a CAN controller and a transceiver. The controller and transceiver are connected through CAN_TX and CAN_RX to transmit logic signals; the transceiver and bus are connected through CAN_High and CAN_Low to transmit differential signals.

24.4.2 Message Structure

Figure 110 Standard Data Frame

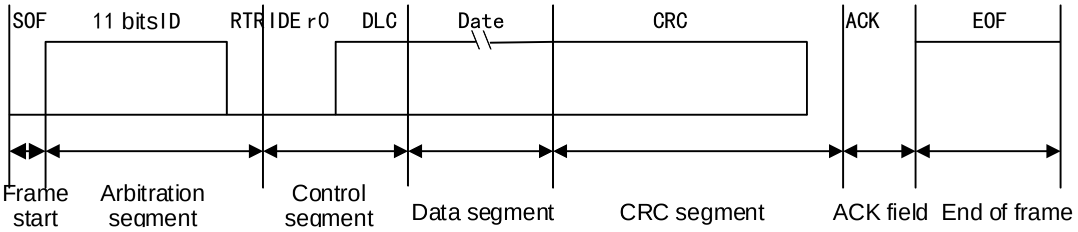
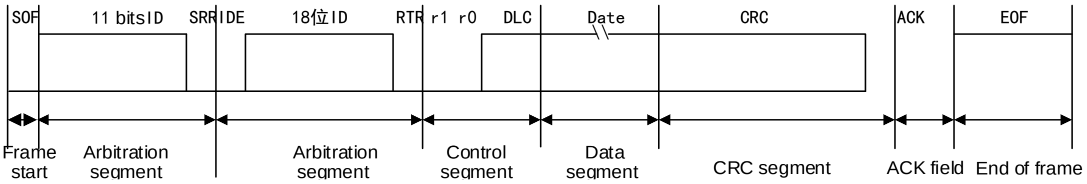


Figure 111 Extended Data Frame



Note:

(1) Frame start: used to inform each node that there will be data for transmission.

(2) Arbitration segment: It is used to decide which message can be transmitted when multiple messages are transmitted. Main content of this segment is ID information, the ID in standard format is 11 bits, and the ID in extended format is 29 bits.

(3) Control segment: The main content of this segment is data length code (DLC), which is used to indicate how many bytes the data segment has in the message. The data segment has up to 8 bytes.

(4) Data segment: Include the data information to be transmitted by the node.

(5) CRC segment: CRC check code is used to ensure correct transmission of the messages.

(6) ACK segment: This segment includes ACK slot bit and ACK delimiter bit. The transmitting node in ACK slot sends recessive bits, while the receiving node sends the dominant bit in this bit to acknowledge.

(7) Frame end: Seven recessive bits transmitted by the transmitting nodes are used to indicate the end.

24.4.3 Working Mode

CAN has three main working modes: initialization mode, normal mode and sleep mode.

24.4.3.1 Initialization mode

Set the INITREQ bit of the configuration register CAN_MCTRL to 1 to request to enter the initialization mode; clear the INITFLG bit to confirm entering the initialization mode.

Clear the INITREQ bit of the configuration register CAN_MCTRL to request to exit the initialization mode; clear the INITFLG bit to confirm exiting the initialization mode.

Message receiving and transmitting is disabled in initialization mode.

24.4.3.2 Normal mode

Clear the INITREQ bit of the configuration register CAN_MCTRL through software to request to enter the normal mode from the initialization mode; wait for the hardware to clear the INITFLG bit to enter the normal mode.

Message receiving and transmitting is allowed in normal mode.

24.4.3.3 Sleep mode

Set the SLEEPREQ bit of the configuration register CAN_MCTRL to 1 to request to enter the sleep mode.

The clock of CAN stops work in sleep mode, the software can normally access the mailbox register, and the CAN is in low-power state.

24.4.4 Communication Mode

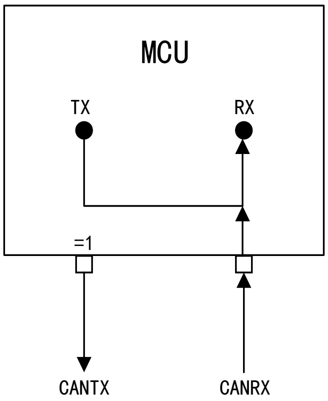
There are four communication modes: silent mode, loopback mode, silent loopback mode and normal mode. Different communication modes can be selected only in initialization mode.

24.4.4.1 Silent mode

Set the SILMEN bit of the configuration register CAN_BITTIM to 1 and select the silent mode.

In this mode, only recessive bit (logic 1)can be transmitted to the bus, while the dominant bit (logic 0) cannot be transmitted, and the data can be received from the bus.

Figure 112 CAN Works in Silent Mode

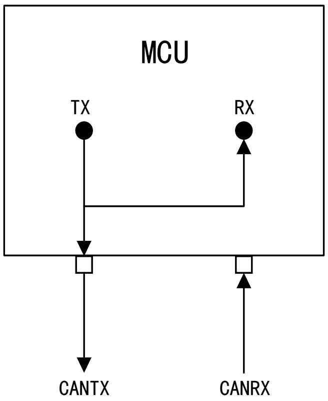


24.4.4.2 Loopback mode

Set the LBKMEN bit of the configuration register CAN_BITTIM to 1 and select the loopback mode.

In this mode, the transmitted data are directly transmitted to the input end for receiving, the data are not received from the bus, and all data can be transmitted to the bus.

Figure 113 CAN Works in Loopback Mode

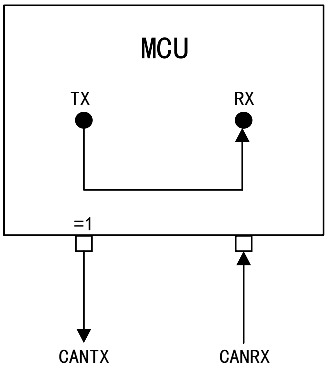


24.4.4.3 Loopback silent mode

Set the LBKMEN and SILMEN bits of the configuration register CAN_BITTIM to 1 and select the loopback silent mode.

In this mode, the transmitted data are directly transmitted to the input end for receiving, and the data are not received from the bus; only recessive bit (logic 1) can be transmitted to the bus, while the dominant bit (logic 0）cannot be transmitted.

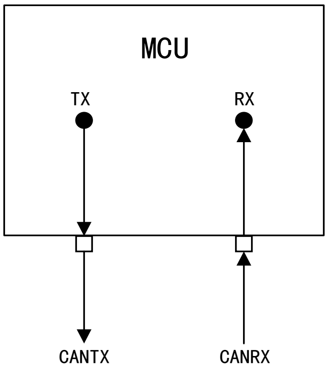
Figure 114 CAN Works in Silent Loopback Mode



24.4.4.4 Normal mode

In this mode, data can be transmitted to the bus and be received from the bus.

Figure 115 CAN Works in Normal Mode



24.4.5 Data Transmission

24.4.5.1 Conversion of transmitting mailbox state

Conversion process of transmitting mailbox state:

（1） First select an empty mailbox to set, submit the transmitting request to the CAN bus controller by setting the TXMREQ bit of the configuration register CAN_TXMIDx to 1, and then the mailbox immediately enters the registration state.

（2） When multiple mailboxes are in the registered state, conduct priority scheduling. When an mailbox has the highest priority, it will enter the predetermined state.

（3） When the message in the transmitting mailbox is transmitted to the bus, it will enter the transmitting state.

（4） After the message is transmitted successfully, the mailbox will become idle again.

24.4.5.2 Transmitting priority

When multiple messages are waiting for transmitting, determine the transmitting sequence through the TXFPCFG bit of the configuration register CAN_MCTRL:

- When the TXFPCFG bit is set to 0, the priority is determined by the message identifier, the identifier is the lowest, the priority is the highest, the identifier is equal, and the message with small mailbox number will be transmitted first
- When the TXFPCFG bit is set to 1, the priority will be determined by the sequence of transmitting request

24.4.5.3 Abort

Transmit the abort request by setting the ABREQFLG bit of the configuration register CAN_TXSTS to 1.

If the mailbox is in registered or predetermined state, stop transmitting the request immediately; if the mailbox is in the transmitting state, there are two conditions: one is that the mailbox is successfully transmitted, the mailbox becomes empty, in such case, the TXSUSFLG bit of the CAN_TXSTS register is set to 1 by hardware; the other is that the mailbox fails to transmit, the mailbox becomes predetermined and the transmitting request is aborted.

24.4.5.4 Automatic retransmission is disabled

Generally, in time triggered communication mode, automatic retransmission should be disabled.

In the mode that the automatic retransmission is disabled, the message is transmitted only once, and no matter what the result is (success, error or arbitration loss), the hardware will not transmit the message again automatically.

When the transmitting process is finished, set the REQCFLG bit of the CAN_TXSTS register to 1, and the transmitting result will be on the TXSUSFLG, ARBLSTFLG and TXERRFLG bits

24.4.6 Data Receiving

24.4.6.1 Receive FIFO

CAN has two receive FIFOs, each FIFO has three mailboxes, the FMNUM[1:0] bit of the register CAN_RXF reflects the number of messages currently stored; set the RFOM bit to1 to release the output mailbox of receive FIFO; FFULLFLG is the full state flag bit; FOVRFLG is overrun state flag bit.

24.4.6.2 Receive FIFO state conversion

At the beginning FIFO is in empty state, and after receiving the message, it will become registered.

When FIFO is in registered state and three mailboxes are full, after receiving next effective message, it will enter the overrun state, and there are two situations for loss of messages in overrun state:

- If FIFO lock function is disabled, the finally received message will be covered by new message

- If FIFO lock function is enabled, the newly received message will be discarded

24.4.7 Filtering Mechanism

Function of the filter: The receiving node decides whether the message is needed according to the message identifier, and only the required message will be received after filtering. CAN controller has 28 filter groups.

24.4.7.1 Bit width

Each group of filters can configure two kinds of bit width.

Figure 116 One 32-bit Filter

ID	CAN_FiBANK1[31:24]	CAN_FiBANK1[23:16]		CAN_FiBANK1[15:8]	CAN_FiBANK1[7:0]			
mapping	STDID[10:3]	STDID[2:0]	EXTID[17:13]	EXTID[12:5]	EXTID[4:0]	IDTYPESEL	TXRFREQ	0

Figure 117 Two 16-bit Filters

ID	CAN_FiBANK1[15:8]	CAN_FiBANK1[7:0]				CAN_FiBANK2[15:8]	CAN_FiBANK2[7:0]			
mapping	STDID[10:3]	STDID [2:0]	TXRF REQ	IDTYP ESEL	EXTID [17:15]	STDID[12:5]	STDID [2:0]	TXRF REQ	IDTYP ESEL	EXTID [17:15]

24.4.7.2 Filtering mode

Mask bit mode

In this mode, it is only required to use some bits of the message identifier as a list to form the mask, and the message ID should be the same as the mask, and then the message can be received

Table 105 Mask Bit Mode Example

ID	1	0	1	1	0	0	1	0	....
Mask	1	0	1	1	1	0	0	1	……
Screened ID	1	X	1	1	0	X	X	0	……

Identifier list mode

In this mode, each bit of the message ID needs to be the same as the filter identifier, and then the message can be received.

Table 106 Identifier List Mode Example

ID	1	1	1	0	1	0	0	1	1
ID	1	1	1	0	1	0	0	1	1
Screened ID	1	1	1	0	1	0	0	1	1

24.4.7.3 Filter priority

The priority rules are as follows:

- The priority of the filter with width of 32 bits is higher than that with width of 16 bits
- Under the condition of the same bit width, the priority of the identifier list mode is higher than that of mask bit mode
- Under the condition of the same bit width and mode, the priority of the small filtering number is high

24.4.8 Bit Timing and Baud Rate

24.4.8.1 Bit timing

The CAN peripheral bit timing of APM32 contains three segments: synchronization segment (SYNC_SEG), time segment 1 (BS1) and time segment 2 (BS2), and the sampling points are at the junction of BS1 and BS2 segments.

- Synchronization segment (SYNC_SEG): This bit occupies one time cell
- Time segment 1 (BS1): This segment occupies one to 16 time cells, and it contains PROP_SEG and PHASE_SEG1 in CAN standard
- Time segment 2 (BS2): This segment occupies one to eight time cells, and it represents PHASE_SEG2 in CAN standard

24.4.8.2 Calculation of baud rate

Time of BS1 segment: T_{s1}=T_q* (TIMSEG1[3:0]+1)

Time of BS2 segment:T_{s2}=T_q* (TIMSEG2[2:0]+1)

Time of one data bit: T1bit=1T_q+$T_{s1+}T_{s2}$

Baud rate=1/ T1bit

T_q =（BRPSC+1）* T_{PCLK}

24.4.9 Error Management

Transmit the error counter through the TXERRCNT bit of the configuration register CAN_ERRSTS and receive the error counter through the RXERRCNT bit of the register CAN_ERRSTS to reflect the error management of CAN bus.

Control the generation of interrupt in error state through the ERRIEN bit of the configuration register CAN_INTEN.

24.4.9.1 Bus-off recovery

When the TXERRCNT of the CAN error state register is greater than 255, the CAN bus controller will enter the bus-off state, then the BOFLG bit of the register CAN_ERRSTS is set to 1, and in this state, the CAN bus controller cannot receive and transmit messages.

Decide the bus-off recovery mode through the ALBOFFM bit of the configuration register CAN_MCTRL:

- If the ALBOFFM bit is set to 1, once the hardware detects 11 continuous recessive bits for 128 times, it will exit the bus-off state automatically;
- If the ALBOFFM bit is set to 0, after the software requests to enter and then exit the initialization mode, it will exit the bus-off state.

24.4.10 Interrupt

Events generating transmitting interrupt:

- The hardware sets REQCFLG0 bit of the register CAN_TXSTS to 1, and the transmitting mailbox 0 becomes idle
- The hardware sets REQCFLG1 bit of the register CAN_TXSTS to 1, and the transmitting mailbox 1 becomes idle
- The hardware sets REQCFLG2 bit of the register CAN_TXSTS to 1, and the transmitting mailbox 2 becomes idle

Events generating FIFO0 interrupt:

- Set the FMNUM0[1:0] bit of the register CAN_RXF0 to a number rather than 0 by the hardware, and FIFO0 will receive a new message
- Set the FFULLFLG0 bit of the register CAN_RXF0 to 1 by the hardware, and FIFO0 will be full
- Set the FOVRFLG0 bit of the register CAN_RXF0 to 1 by the hardware and FIFO0 will overrun

Events generating FIFO1 interrupt:

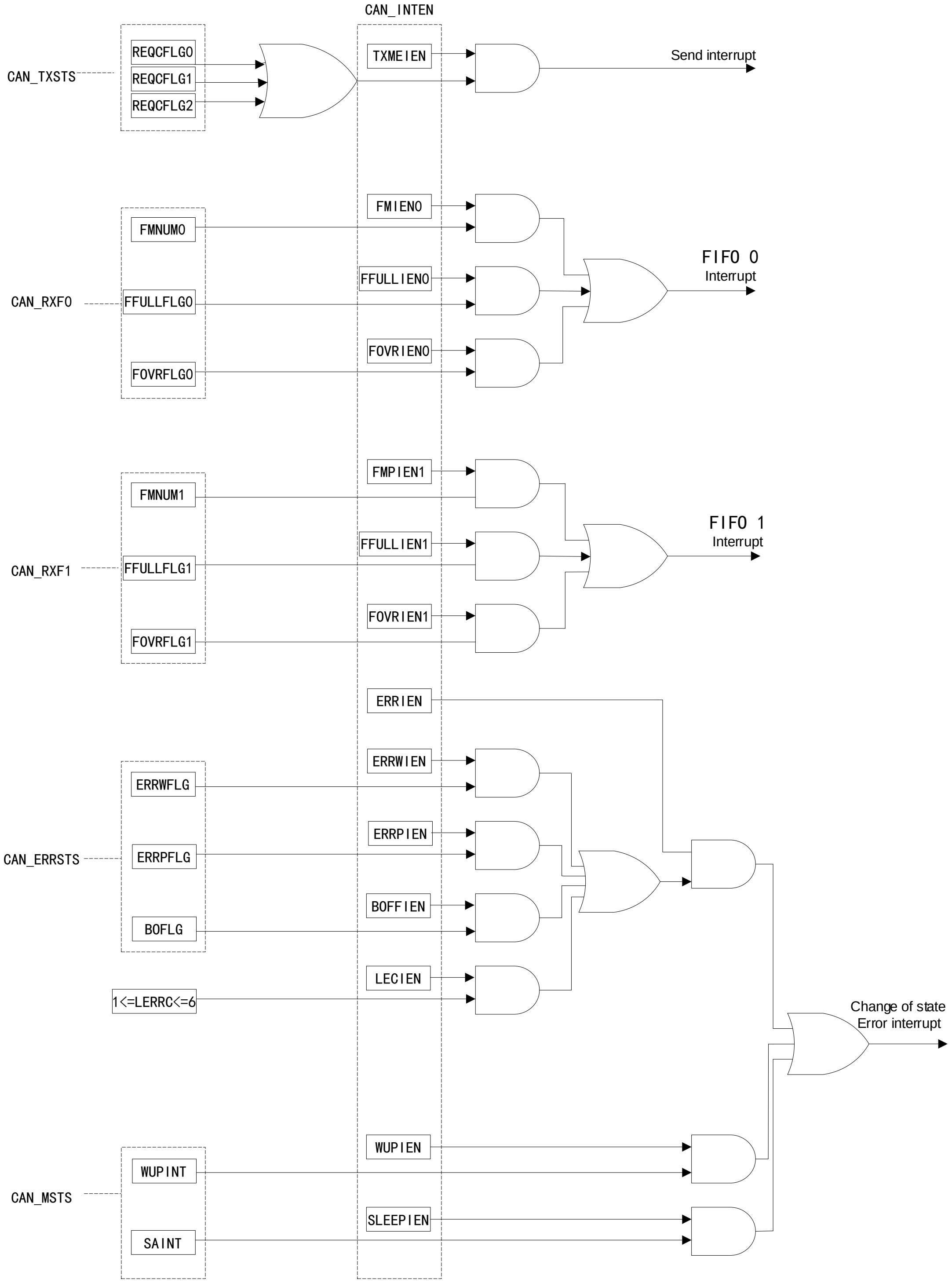
- Set the FMNUM1[1:0] bit of the register CAN_RXF1 to a number rather than 0 by the hardware, and FIFO1 will receive a new message
- Set the FFULLFLG1 bit of the register CAN_RXF1 to 1 by the hardware, and FIFO1 will be full
- Set the FOVRFLG1 bit of the register CAN_RXF1 to 1 by the hardware and FIFO1 will overrun

Events generating state change and error interrupt:

- Set the SLEEPIEN bit of the register CAN_INTEN to 1 by the hardware and it will enter the sleep mode
- Set the WUPIEN bit of the register CAN_INTEN to 1 by the hardware and interrupt enable will be woken up
- Set the ERRWFLG bit of the register CAN_ERRSTS to 1 by the hardware, and it means that the number of errors has reached the threshold
- Set the ERRPFLG bit of the register CAN_ERRSTS to 1 by the hardware, and it means that the number of errors has reached the threshold of passive error

- Set the LERRC[2:0] bit of the register CAN_ERRSTS by the hardware, and it indicates the condition of last error

Figure 118 Event Flag and Interrupt Generation



24.5 Register Address Mapping

CAN1 base address: 0x4000_6400

CAN2 base address: 0x4000_6800

Note: Except base address, the register and offset addresses of CAN1 and CAN2 are exactly the same.

Table 107 CAN Register Address Mapping

Register name	Description	Offset address
CAN_MCTRL	CAN main control register	0x00
CAN_MSTS	CAN main state register	0x04
CAN_TXSTS	CAN transmit state register	0x08
CAN_RXF0	CAN receive FIFO 0 register	0x0C
CAN_RXF1	CAN receive FIFO 1 register	0x10
CAN_INTEN	CAN interrupt enable register	0x14
CAN_ERRSTS	CAN error state register	0x18
CAN_BITTIM	CAN bit timing register	0x1C
CAN_TXMIDx	Transmitting mailbox identifier register	0x180, 0x190, 0x1A0
CAN_TXDLENx	Transmitting mailbox data length register	0x184, 0x194, 0x1A4
CAN_TXMDLx	Transmitting mailbox low-byte data register	0x188, 0x198, 0x1A8
CAN_TXMDHx	Transmitting mailbox high-byte data register	0x18C, 0x19C, 0x1AC
CAN_RXMIDx	Receive FIFO mailbox identifier register	0x1B0, 0x1C0
CAN_RXDLENx	Receive FIFO mailbox data length register	0x1B4, 0x1C4
CAN_RXMDLx	Receive FIFO mailbox low-byte data register	0x1B8, 0x1C8
CAN_RXMDHx	Receive FIFO mailbox high-byte data register	0x1BC, 0x1CC
CAN_FCTRL	CAN filter main control register	0x200
CAN_FMCFG	CAN filter mode register	0x204
CAN_FSCFG	CAN filter bit width register	0x20C
CAN_FFASS	CAN filter FIFO association register	0x214
CAN_FACT	CAN filter activation register	0x21C
CAN_FiBANKx	Register x of CAN filter group i	0x240..0x2AC

24.6 Register Functional Description

24.6.1 CAN control and state register

24.6.1.1 CAN main control register (CAN_MCTRL)

Offset address: 0x00

Reset value: 0x0001 0002

Field	Name	R/W	Description
0	INITREQ	R/W	Request to Enter Initialization Mode

Field	Name	R/W	Description
			0: Enter the normal work mode from the initialization mode 1: Enter the initialization mode from the normal work mode
1	SLEEPREQ	R/W	Request to Enter Sleep Mode 0: Exit the sleep mode 1: Request to enter the sleep mode. If the AWUPCFG bit is set to 1, when the RX signal detects CAN message, this bit will be cleared by hardware; after reset, reset this bit to 1; after reset, it will enter the sleep mode.
2	TXFPCFG	R/W	Transmit FIFO Priority Configure This bit is used to determine which parameters determine the transmission priority when multiple messages are waiting for transmission. 0: Determined by the message identifier 1: Determined by the sequence of transmission request
3	RXFLOCK	R/W	Receive FIFO Locked Mode Configure This bit is used to determine whether FIFO is locked when receiving overrun, and how to deal with the next received message when the message of the receive FIFO has not been read out. 0: Unlocked; If the message of the receive FIFO is not read out, the next received message will cover the original message 1: Locked; when the message of the received FIFO is not read out, the next received message will be discarded
4	ARTXMD	R/W	Automatic Retransmission Message Disable 0: Automatic retransmission is enabled, and the message will be retransmitted automatically until it is transmitted successfully 1: Automatic retransmission is disabled and the message is transmitted only once
5	AWUPCFG	R/W	Automatic Wakeup Mode Configure 0: Software wakes up the sleep mode by clearing the SMREQ bit of the CAN_MCTRL register 1: Hardware wakes up the sleep mode by detecting CAN message
6	ALBOFFM	R/W	Automatic Leaving Bus-Off Status Condition Management 0: After the software resets the INITREQ bit of the CAN_MCTRL register to 1 and then clears it, when the hardware detects 11 continuous recessive bits for 128 times, it will exit from the bus-off state 1: When the hardware detects 11 continuous recessive bits for 128 times, it will exit from the bus-off state automatically
14:7	Reserved		
15	SWRST	R/S	Software Reset CAN 0: Work normally 1: CAN is reset by force, and after reset, CAN enters the sleep mode; the hardware will clear this bit automatically
16	DBGFRZE	R/W	Debug Freeze 0: Invalid 1: During debugging, CAN cannot receive/transmit, but it still can read and write and control the receive FIFO normally
31:17	Reserved		

24.6.1.2 CAN main state register (CAN_MSTS)

Offset address: 0x04

Reset value: 0x0000 0C02

Field	Name	R/W	Description
0	INITFLG	R	Being Initialization Mode Flag This bit is set to 1 or cleared by hardware. 1: Exit the initialization mode 1: Being in the initialization mode; this bit is confirmation for initialization request bit of the CAN_MCTRL register.
1	SLEEPFLG	R	Being Sleep Mode Flag This bit is set to 1 or cleared by hardware 0: Exit the sleep mode 1: Being in the sleep mode; this bit is confirmation for sleep moderequest bit of the CAN_MCTRL register.
2	ERRIFLG	RC_W1	Error Interrupt Occur Flag This bit is set to 1 by hardware and written to 1 and cleared by software. 0: Not occur 1: Occurred
3	WUPIFLG	RC_W1	Wakeup Interrupt Occur Flag When entering the sleep mode and detecting SOP wake-up, the bit is set to 1 by hardware; it is written to 1 and cleared by software. 0: Failed to wake up from the sleep mode 1: Woke up from the sleep mode
4	SLEEPIFLG	RC_W1	Being Sleep Mode Interrupt Flag When entering the sleep mode, this bit is set to 1 by hardware and corresponding interrupt will be triggered; when exiting the sleep mode, this bit is cleared by hardware and is written as 1 and cleared by software. 0: Failed to enter the sleep mode 1: Entered the sleep mode
7:5	Reserved		
8	TXMFLG	R	Being Transmit Mode Flag 0: CAN is not in transmission mode 1: CAN is in transmission mode
9	RXMFLG	R	Being Receive Mode Flag 0: CAN is not in receiving mode 1: CAN is in receiving mode
10	LSAMVALUE	R	CAN Rx Pin Last Sample Value
11	RXSIGL	R	CAN Rx Pin Signal Level
31:12	Reserved		

24.6.1.3 CAN transmit state register (CAN_TXSTS)

Offset address: 0x08

Reset value: 0x1C00 0000

Field	Name	R/W	Description
0	REQCFLG0	RC_W1	Mailbox 0 Request Completed Flag When the last transmission or abortion request of the mailbox 0 is completed, this bit is set to 1 by hardware; when receiving the transmission request, this bit is cleared by hardware; it is written to 1 or cleared by software. 0: Being transmitted 1: Transmission completed
1	TXSUSFLG0	RC_W1	Mailbox 0 Transmission Success Flag When mailbox 0 attempts to transmit successfully, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Last transmission attempt failed 1: Last transmission attempt succeeded
2	ARBLSTFLG0	RC_W1	Mailbox 0 Arbitration Lost Flag When the mailbox 0 loses arbitration, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Meaningless 1: Lost
3	TXERRFLG0	RC_W1	Mailbox 0 Transmission Error Flag When mailbox 0 fails to transmit, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Meaningless 1: Failed to transmit
6:4	Reserved		
7	ABREQFLG0	R/S	Mailbox 0 Abort Request Flag If there is no message waiting for transmitting in mailbox 0, this bit is ineffective. 0: The transmitting message of mailbox 0 is cleared, and this bit is cleared by hardware 1: Set this bit to 1 to abort the transmission request of mailbox 0
8	REQCFLG1	RC_W1	Mailbox 1 Request Completed Flag When the last request of mailbox 1 is transmitted or aborted, this bit is set to 1 by hardware; When receiving the transmission request, this bit is cleared by hardware, and written to 1 and cleared by software. 0: Being transmitted 1: Transmission completed
9	TXSUSFLG1	RC_W1	Mailbox 1 Transmission Success Flag When mailbox 1 attempts to transmit successfully, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Last transmission attempt failed 1: Last transmission attempt succeeded
10	ARBLSTFLG1	RC_W1	Mailbox 1 Arbitration Lost Flag When the mailbox 1 loses arbitration, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Meaningless 1: Lost

Field	Name	R/W	Description
11	TXERRFLG1	RC_W1	Mailbox 1 Transmission Error Flag When mailbox 1 fails to transmit, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Meaningless 1: Failed to transmit
14:12	Reserved		
15	ABREQFLG1	R/S	Mailbox 1 Abort Request Flag If there is no message waiting for transmitting in mailbox 1, this bit is ineffective. 0: The transmitting message of mailbox 1 is cleared, and this bit is cleared by hardware 1: Set this bit to 1 to abort the transmission request of mailbox 1
16	REQCFLG2	RC_W1	Mailbox 2 Request Completed Flag When the last transmission or abortion request of the mailbox 2 is completed, this bit is set to 1 by hardware; when receiving the transmission request, this bit is cleared by hardware; it is written to 1 or cleared by software. 0: Being transmitted 1: Transmission completed
17	TXSUSFLG2	RC_W1	Mailbox 2 Transmission Success Flag When mailbox 2 attempts to transmit successfully, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Last transmission attempt failed 1: Last transmission attempt succeeded
18	ARBLSTFLG2	RC_W1	Mailbox 2 Arbitration Lost Flag When the mailbox 2 loses arbitration, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Meaningless 1: Lost
19	TXERRFLG2	RC_W1	Mailbox 2 Transmission Error Flag When mailbox 2 fails to transmit, this bit is set to 1 by hardware; and written to 1 and cleared by software. 0: Meaningless 1: Failed to transmit
22:20	Reserved		
23	ABREQFLG2	R/S	Mailbox 2 Abort Request Flag If there is no message waiting for transmitting in mailbox 2, this bit is ineffective. 0: The transmitting message of mailbox 2 is cleared, and this bit is cleared by hardware 1: Set this bit to 1 to abort the transmission request of mailbox 2
25:24	EMNUM[1:0]	R	Empty Mailbox Number This bit is applicable when there is empty mailbox. When all the transmitting mailboxes are empty, it means the number of the transmitting mailbox with the lowest priority; when the mailbox is not empty but not all empty, it means the number of next mailbox to be transmitted.

Field	Name	R/W	Description
26	TXMEFLG0	R	Transmit Mailbox 0 Empty Flag When the transmitting mailbox 0 is empty, this bit is set to 1 by hardware. 0: There is message to be transmitted in mailbox 0 1: There is no message to be transmitted in mailbox 0
27	TXMEFLG1	R	Transmit Mailbox 1 Empty Flag When the transmitting mailbox 1 is empty, this bit is set to 1 by hardware. 0: There is message to be transmitted in mailbox 1 1: There is no message to be transmitted in mailbox 1
28	TXMEFLG2	R	Transmit Mailbox 2 Empty Flag When the transmitting mailbox 2 is empty, this bit is set to 1 by hardware. 0: There is message to be transmitted in mailbox 2 1: There is no message to be transmitted in mailbox 2
29	LOWESTP0	R	the Lowest Transmission Priority Flag For Mailbox 0 0: Meaningless 1: The priority of mailbox 0 is the lowest among those mailboxes waiting to transmit messages Note: If there is only one mailbox waiting, LOWESTP[2:0] is cleared.
30	LOWESTP1	R	the Lowest Transmission Priority Flag For Mailbox 1 0: Meaningless 1: The priority of mailbox 1 is the lowest among those mailboxes waiting to transmit messages
31	LOWESTP2	R	the Lowest Transmission Priority Flag For Mailbox 2 0: Meaningless 1: The priority of mailbox 2 is the lowest among those mailboxes waiting to transmit messages

24.6.1.4 CAN receive FIFO 0 register (CAN_RXF0)

Offset address: 0x0C

Reset value: 0x00

Field	Name	R/W	Description
1:0	FMNUM0[1:0]	R	the number of Message in receive FIFO0 These bits are used to reflect the number of messages stored in current receive FIFO0. Every time a new message is received, add 1 to FMNUM0 bit; every time the mailbox message is released and outputted, subtract 1 from FMNUM0 bit.
2	Reserved		
3	FFULLFLG0	RC_W1	Receive FIFO0 Full Flag When there are three messages in FIFO0, it means the FIFO0 has been full; this bit is set to 1 by hardware and written to 1 and cleared by software. 0: Not full 1: Full

Field	Name	R/W	Description
4	FOVRFLG0	RC_W1	Receive FIFO 0 Overrun Flag When there are three messages in FIFO0 and then a new message is received, it means the FIFO0 overrun; this bit is set to 1 by hardware and written to 1 and cleared by software. 0: No overrun 1: Overrun is generated
5	RFOM0	R/S	Release Receive FIFO0 Output Mailbox to Receive Massage This bit is set to 1 by hardware and cleared by software. If there is no message in FIFO, this bit is invalid. When FIFO contains more than two messges, the output mailbox must be first released to acess the second message. 0: Meaningless 1: Release the output mailbox of receive FIFO0
31:6	Reserved		

24.6.1.5 CAN receive FIFO 1 register (CAN_RXF1)

Offset address: 0x10

Reset value: 0x00

Field	Name	R/W	Description
1:0	FMNUM1[1:0]	R	the number of Message in receive FIFO1 These bits are used to reflect the number of messages stored in current receive FIFO1. Every time a new message is received, add 1 to FMNUM1 bit; every time the mailbox message is released and outputted, subtract 1 from FMNUM1 bit.
2	Reserved		
3	FFULLFLG1	RC_W1	Receive FIFO0 Full Flag When there are three messages in FIFO1, it means the FIFO1 has been full; this bit is set to 1 by hardware and written to 1 and cleared by software. 0: Not full 1: Full
4	FOVRFLG1	RC_W1	Receive FIFO1 Overrun Flag When there are three messages in FIFO1 and then a new message is received, it means the FIFO1 overrun ; this bit is set to 1 by hardware and written to 1 and cleared by software. 0: No overrun 1: Overrun is generated
5	RFOM1	R/S	Release Receive FIFO1 Output Mailbox to Receive Massage This bit is set to 1 by hardware and cleared by software. If there is no message in FIFO, this bit is invalid. When FIFO contains more than two messges, the output mailbox must be first released to acess the second message. 0: Meaningless 1: Release the output mailbox of receive FIFO1
31:6	Reserved		

24.6.1.6 CAN interrupt enable register (CAN_INTEN)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	TXMEIEN	R/W	Transmit Mailbox Empty Interrupt Enable When REQCFLGx bit is set to 1, it means transmission has been completed, and the transmitting mailbox is empty; if this bit is set to 1, an interrupt will be generated. 0: No interrupt 1: Interrupt generated
1	FMIEN0	R/W	Interrupt Enable When The Number Of FIFO0 Message Is Not 0 When FMNUM0[1:0] bit of FIFO 0 is not zero, it means that the number of messages in FIFO0 is not zero; if this bit is set to 1, an interrupt will be generated. 0: No interrupt 1: Interrupt generated
2	FFULLIEN0	R/W	FIFO0 Full Interrupt Enable When the FFULLFLG0 bit of FIFO0 is set to 1, it means that the message of FIFO0 is full; if this bit is set to 1, an interrupt will be generated. 0: No interrupt 1: Interrupt generated
3	FOVRIEN0	R/W	FIFO0 Overrun Interrupt Enable When the FOVRFLG0 bit of FIFO0 is set to 1, it means that the FIFO0 has been overloaded; if this bit is set to 1, an interrupt will be generated. 0: No interrupt 1: Interrupt generated
4	FMPIEN1	R/W	I Interrupt Enable wnhen the number of FIFO1 Message is not 0 When FMNUM1[1:0] bit of FIFO 1 is not zero, it means that the number of messages in FIFO1 is not zero; if this bit is set to 1, an interrupt will be generated. 0: No interrupt 1: Interrupt generated
5	FFULLIEN1	R/W	FIFO1 Full Interrupt Enable When the FFULLFLG1 bit of FIFO1 is set to 1, it means that the message of FIFO1 is full; if this bit is set to 1, an interrupt will be generated. 0: No interrupt 1: Interrupt generated
6	FOVRIEN1	R/W	FIFO1 Overrun Interrupt Enable When the FOVRFLG1 bit of FIFO1 is set to 1, it means that the FIFO1 has been overloaded; if this bit is set to 1, an interrupt will be generated. 0: No interrupt 1: Interrupt generated
7	Reserved		

Field	Name	R/W	Description
8	ERRWIEN	R/W	Error Warning Interrupt Enable When ERRWFLG bit is set to 1, an error warning will occur; if this bit is set to 1, ERRIFLG shall be set and a warning error interrupt will be generated. 0: ERRIFLG bit is not set 1: ERRIFLG bit is set to 1
9	ERRPIEN	R/W	Error Passive Interrupt Enable When ERRPFLG bit is set to 1, a pssive error will occur; if this bit is set to 1, ERRIFLG shall be set and a passive error interrupt will be generated. 0: ERRIFLG bit is not set 1: ERRIFLG bit is set to 1
10	BOFFIEN	R/W	Bus-Off Interrupt Enable When BOFFFLG bit is set to 1, bus-off will occur; if this bit is set to 1, ERRIFLG shall be set and an bus-off error interrupt will be generated. 0: ERRIFLG bit is not set 1: ERRIFLG bit is set to 1
11	LECIEN	R/W	Last Error Code Interrupt Enable When an error is detected and the hardware sets LERRC[2:0], the last error code is recorded. If this bit set to 1, the ERRIFLG is set to generate the last error interrupt. 0: ERRIFLG bit is not set 1: ERRIFLG bit is set to 1
14:12	Reserved		
15	ERRIEN	R/W	Error interrupt Enable When the corresponding error state register is set to 1, if this bit is set to 1, an error interrupt will be generated. 0: No interrupt 1: Interrupt generated
16	WUPIEN	R/W	Wakeup Interrupt Enable When WUPINT bit is set to 1, if this bit is set to 1, a wake-up interrupt will be generated. 0: No interrupt 1: Interrupt generated
17	SLEEPIEN	R/W	Sleep Interrupt Enable When SLEEPIFLG bit is set to 1, if this bit is set to 1, a sleep interrupt will be generated. 0: No interrupt 1: Interrupt generated
31:18	Reserved		

24.6.1.7 CAN error state register (CAN_ERRSTS)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ERRWFLG	R	Error Warning Occur Flag When the value of the receiving error counter or transmitting error counter ≥96, this bit is set to 1 by hardware. 0: No error warning 1: Error warning occurred
1	ERRPFLG	R	Error Passive Occur Flag When the value of the receiving error counter or transmitting error counter ≥127, this bit is set to 1 by hardware. 0: No passive error 1: Passive error appears
2	BOFLG	R	Enter Bus-Off Flag When the value of the transmitting error counter TXERRCNT is greater than 255, CAN will enter the bus-off state and this bit is set to 1 by hardware. 0: CAN not in bus-off state 1: CAN in bus-off state
3	Reserved		
6:4	LERRC	R/W	Record Last Error Code When the error on CAN bus is detected, it is set by hardware according to the error category; when the message is transmitted or received correctly, this bit is cleared by hardware. 000: No error 001: Bit stuffing error 010: Form (Form) error 011: Acknowledgment (ACK) error 100: Recessive bit error 101: Dominant bit error 110: CRC error 111: Set by software
15:7	Reserved		
23:16	TXERRCNT	R	Least Significant Byte Of The 9-Bit Transmit Error Counter The counter is implemented according to the transmission part of fault definition mechanism of CAN protocol.
31:24	RXERRCNT	R	Receive Error Counter The receiving error counter is implemented according to the receiving part of fault definition mechanism of CAN protocol. When receiving error occurs, according to the condition of error, add 1 or 8 to the counter, and subtract 1 after receiving successfully. When the value of the counter is greater than 127, set the counter value to 120.

24.6.1.8 CAN bit timing register (CAN_BITTIM)

Offset address: 0x1C

Reset value: 0x0123 0000

Field	Name	R/W	Description
9:0	BRPSC	R/W	Baud Rate Prescaler Factor Setup Time cell t_q =(BRPSC+1)× t_{PCLK}

15:10	Reserved		
19:16	TIMSEG1	R/W	Time Segment 1 Setup Time occupied by time period 1 t_{BS1} = t_{CAN} x (TIMSEG1+1).
22:20	TIMSEG2	R/W	Time Segment 2 Setup Time occupied by time period 2 t_{BS2} = t_{CAN} x (TIMSEG2+1).
23	Reserved		
25:24	RSYNJW	R/W	Resynchronization Jump Width Time that CAN hardware can extend or shorten in this bit t_{RJW}=t_{CAN} x (RSYNJW+1).
29:26	Reserved		
30	LBKMEN	R/W	Loop Back Mode Enable 0: Disable 1: Enable
31	SILMEN	R/W	Silent Mode Enable 0: Normal state 1: Silent mode

Note: When CAN is in initialization mode, this register can be accessed only by software

24.6.2 CAN mailbox register

This section describes the transmitting and receiving mailbox registers.

The transmitting and receiving mailboxes are almost the same except the following examples:

- FMIDX domain of CAN_RXDLENx register;
- The receiving mailbox is read-only;
- The transmitting mailbox is writable only when it is empty, and if the corresponding TXMEFLG bit of CAN_TXSTS register is 1, it means the transmitting mailbox is empty.

There are three transmitting mailboxes and two receiving mailboxes in total. Each receiving mailbox is FIFO with three levels of depth, and can only access the message that is received first in FIFO.

24.6.2.1 Transmitting mailbox identifier register (CAN_TXMIDx) (x=0..2)

Offset address: 0x180, 0x190, 0x1A0

Reset value: 0xXXXX XXXX, X=undefined bit (except Bit 0, TXMREQ=0 after reset)

Field	**Name**	**R/W**	**Description**
0	TXMREQ	R/W	Transmit Mailbox Data Request 0: When the data in the mailbox is transmitted, the mailbox is empty and this bit is cleared by hardware 1: Software writes 1, to enable request to transmit mailbox data
1	TXRFREQ	R/W	Transmit Remote Frame Request 0: Data frame 1: Remote frame

Field	Name	R/W	Description
2	IDTYPESEL	R/W	Identifier Type Select 0: Stanard identifier 1: Extended identifier
20:3	EXTID[17:0]	R/W	Extended Identifier Setup Low byte of extended identity label.
31:21	STDID[10:0]/EXTID[28:18]	R/W	Standard Identifier Or Extended Identifier According to the content of IDTYPESEL bit, these bits are standard identifier STDID[10:0] and high byte EXTID[28:18] of extended identifier.

Note: 1. When its mailbox is in the state of waiting for transmission, this register is write-protection

2. This register realizes transmission request control function (No. 0 bit) - the reset value is 0

24.6.2.2 Transmitting mailbox data length register (CAN_TXDLENx) (x=0..2)

When the mailbox is not idle, all bits of this register are write-protected.

Offset address: 0x184, 0x194, 0x1A4

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
3:0	DLCODE	R/W	Transmit Data Length Code Setup
31:4	Reserved		

24.6.2.3 Transmitting mailbox low-byte data register (CAN_TXMDLx) (x=0..2)

When the mailbox is not idle, all bits of this register are write-protected, and the message contains 0 to 7-byte data and starts from the byte 0.

Offset address: 0x188, 0x198, 0x1A8

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
7:0	DATABYTE0	R/W	Data Byte 0 of the Message
15:8	DATABYTE1	R/W	Data Byte 1 of the Message
23:16	DATABYTE2	R/W	Data Byte 2 of the Message
31:24	DATABYTE3	R/W	Data Byte 3 of the Message

24.6.2.4 Transmitting mailbox high-byte data register (CAN_TXMDHx) (x=0..2)

When the mailbox is not idle, all bits of this register are write-protected.

Offset address: 0x18C, 0x19C, 0x1AC

Reset value: 0xXXXX XXXX, X=undefined bit

Field	Name	R/W	Description
7:0	DATABYTE4	R/W	Data Byte 4 of the Message
15:8	DATABYTE5	R/W	Data Byte 5 of the Message
23:16	DATABYTE6	R/W	Data Byte 6 of the Message
31:24	DATABYTE7	R/W	Data Byte 7 of the Message

24.6.2.5 Receive FIFO mailbox identifier register (CAN_RXMIDx) (x=0..1)

Offset address: 0x1B0, 0x1C0

Reset value: 0xXXXX XXXX, X=undefined bit

Field	Name	R/W	Description
0	Reserved		
1	RFTXREQ	R	Remote Frame Transmission Request 0: Data frame 1: Remote frame
2	IDTYPESEL	R	Identifier Type Select 0: Stanard identifier 1: Extended identifier
20:3	EXTID[17:0]	R	Extended Identifier Setup Low byte of extended identifier.
31:21	STDID[10:0]/EXTID[28:18]	R	Standard Identifier Or Extended Identifier According to the content of IDTYPESEL bit, these bits are standard identifier STDID[10:0] and high byte EXTID[28:18] of extended identifier.

Note: All receiving mailbox registers are read-only.

24.6.2.6 Receive FIFO mailbox data length register (CAN_RXDLENx) (x=0..1)

Offset address: 0x1B4, 0x1C4

Reset value: 0xXXXXX XXXX

Field	Name	R/W	Description
3:0	DLCODE	R	Receive Data Length Code Setup This bit represents the data length in the frame; for remote frame, DLCODE is constantly 0.
7:4	Reserved		
15:8	FMIDX	R	Filter Match Index Setup
31:16	Reserved		

Note: All receiving mailbox registers are read-only.

24.6.2.7 Receive FIFO mailbox low-byte data register (CAN_RXMDLx) (x=0..1)

Offset address: 0x1B8, 0x1C8; the message contains 0 to 8-byte data, which starts from the byte 0.

Reset value: 0xXXXXX XXXX

Field	Name	R/W	Description
7:0	DATABYTE0	R	Data Byte 0 of the Message
15:8	DATABYTE1	R	Data Byte 0 of the Message
23:16	DATABYTE2	R	Data Byte 0 of the Message
31:24	DATABYTE3	R	Data Byte 0 of the Message

Note: All receiving mailbox registers are read-only.

24.6.2.8 Receive FIFO mailbox high-byte data register (CAN_RXMDHx) (x=0..1)

Offset address: 0x1BC, 0x1CC

Reset value: 0xXXXX XXXX, X=undefined bit

Field	Name	R/W	Description
7:0	DATABYTE4	R	Data Byte 0 of the Message
15:8	DATABYTE5	R	Data Byte 0 of the Message
23:16	DATABYTE6	R	Data Byte 0 of the Message
31:24	DATABYTE7	R	Data Byte 0 of the Message

Note: All receiving mailbox registers are read-only.

24.6.3 CAN filter register

24.6.3.1 CAN filter main control register (CAN_FCTRL)

Offset address: 0x200

Reset value: 0x2A1C 0E01

Field	Name	R/W	Description
0	FINITEN	R/W	Filter Init Mode Enable 0: Normal mode 1: Initialization mode
7:1	Reserved		
13:8	CAN2SB	R/W	CAN2 Start Bank They define the start bank for the CAN2 interface (Slave) in the range 0 to 27. Note:When CAN2SB[5:0] = 28d, all the filters to CAN1 can be used. When CAN2SB[5:0] is set to 0, all the filters to CAN2 can be used.
31:14	Reserved		

Note: The non-reserved bit of this register is completely controlled by software.

24.6.3.2 CAN filter mode configuration register (CAN_FMCFG)

Offset addres: 0x204

Reset value: 0x0000 0000

Field	Name	R/W	Description
27:0	FMCFGx	R/W	Filter Mode Configure The value of x is within 0-27. 0: Identifier mask bit mode 1: Identifier list mode
31:28	Reserved		

Note: Only when CAN_FCTRL (FINITEN =1) is set to make the filter in initialization mode, can this register be written.

24.6.3.3 CAN filter bit width register (CAN_FSCFG)

Offset address: 0x20C

Reset value: 0x0000 0000

Field	Name	R/W	Description
27:0	FSCFGx	R/W	Filterx Scale Configure The value of x is within 0-27. 0: Two 16 bits 1: Single 32 bits
31:28	Reserved		

Note: Only when CAN_FCTRL (FINITEN =1) is set to make the filter in initialization mode, can this register be written.

24.6.3.4 CAN filter FIFO association register (CAN_FFASS)

Offset address: 0x214

Reset value: 0x0000 0000

Field	Name	R/W	Description
27:0	FFASSx	R/W	Configure Filterx Associated with FIFO The value of x is within 0-27. 0: The filter is associted with FIFO0 1: The filter is associted with FIFO1
31:28	Reserved		

Note: Only when CAN_FCTRL (FINITEN =1) is set to make the filter in initialization mode, can this register be written.

24.6.3.5 CAN filter activation register (CAN_FACT)

Offset address: 0x21C

Reset value: 0x0000 0000

Field	Name	R/W	Description
27:0	FACTx	R/W	Filterx Active The value of x is within 0-27. 0: Disable 1: Active
31:28	Reserved		

24.6.3.6 Register x of CAN filter group x (CAN_FiBANKx) (i = 0..27； x=1..2)

Offset address:0x240..0x31C

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
31:0	FBIT[31:0]	R/W	Filter Bits Setup Identifier list mode: 0: FBITx bit is dominant bit 1: FBITx bit is recessive bit Identifier mask bit mode: 0: FBITx is not used for comparison 1: FBITx must match Note: The value of x is 0~31, indicating the bit number of FBIT.

Note: There are 28 sets of filters in product: i=0..27. Each set of filters consists of two 32-bit registers and CAN_FiBANK[2:1]. The corresponding filter registers can be modified only when the corresponding FACTx bit of CAN_FACT register is cleared or the FINITEN bit of CAN_FCTRL register is 1.

25 Secure Digital Input/Output Interface (SDIO)

25.1 Full Name and Abbreviation Description of Terms

Table 108 Full Name and Abbreviation Description of Terms

Full name in English	English abbreviation
First Input First Output	FIFO
Command Path State Machine	CPSM
Data Path State Machine	DPSM

25.2 Introduction

The secure digital input/output interface can connect SD card, SD I/O card, multi-media card (MMC) and CE-ATA card master interfaces, and provide data transmission between AHB system bus and SD memory card, SD I/O card, MMC and CE-ATA device.

25.3 Main Characteristics

（1） SD card: Compatible with SD memory card specification version 2.0

（2） SD I/O card: Compatible with SD I/O card specification version 2.0: support two different bus modes: 1 bit (default) and four bits.

（3） MMC: Compatible with multimedia card system specification 4.2 and previous versions. Three different data bus modes are supported:1 bit (default), 4 bits and 8 bits.

（4） CE-ATA: Compatible with CE-ATA digital protocol version 1.1.

（5） The data transmission rate in 8-bit bus mode is up to 48MHz

（6） Interrupt and DMA request

（7） Data and command output enable signal, used for controlling bidirectional driver.

Note: SDIO of current version only supports one SD/SD IO/MMC 4.2 card at the same time, however, it supports multiple MMC4.1 or cards of previous version.

25.4 Functional Description

SDIO structure mainly contains two parts:

SDIO adapter: Realize the related functions of MMC/SD/SD I/O cards, and consist of control unit, data unit and command unit. The control unit manages the clock signal, the data unit manages the data transmission, and the command unit manages the command transmission.

AHB bus interface: Operate the registers in SDIO adapter, used for FIFO unit for data transmission, generate an interrupt and DMA request signal.

Figure 119 SDIO Structure Block Diagram

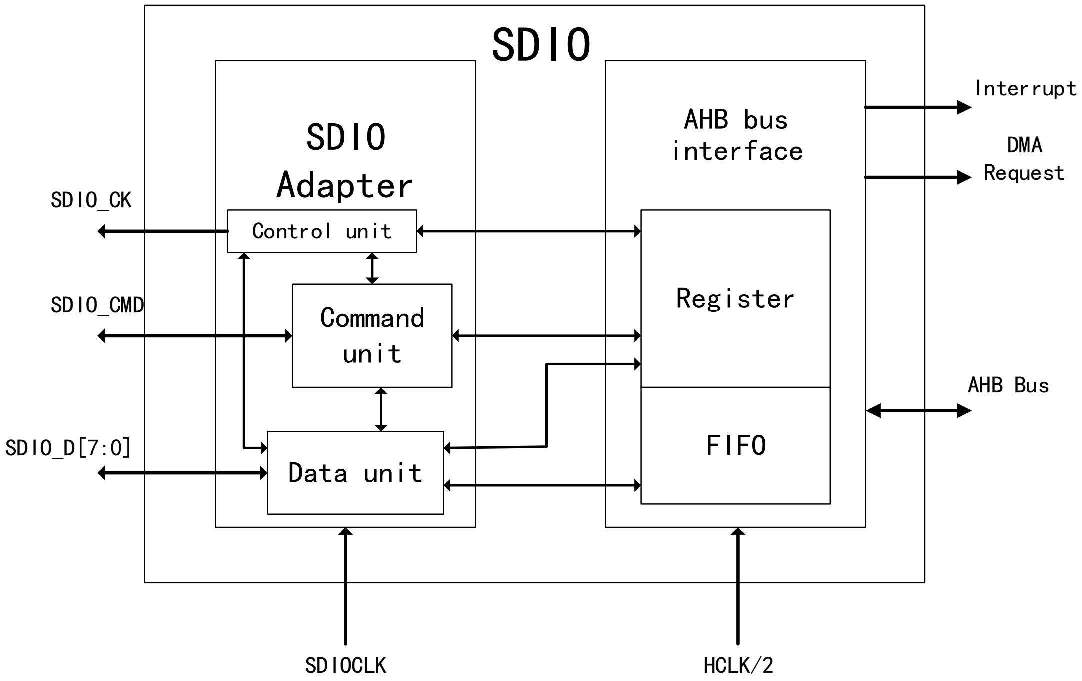


Table 109 SDIO Pin Definition

Pin	Direction	Instruction
SDIO_CK	Output	MMC/SD/SD I/O card clock, clock line from master to card
SDIO_CMD	Bilateral	MMC/SD/SD I/O card command, bidirectional command signal
SDIO_D[7:0]	Bilateral	MMC/SD/SD I/O card data, bidirectional data bus

25.4.1 SDIO Bus Topology

After power-on reset, the master must initialize the device through a special message-based bus protocol.

Each message is represented by one of the following parts:

- Command: Command is a token to start an operation, from the master to the card, and the command is transmitted to the CMD line serially.
- Response: From the card to the master, as a response to the previously received command, the response is transmitted onto the CMD serially.
- Data: It can be transmitted from the master to the card or from the card to the master. Transmit through data cable. The number of data cables for data transmission can be 1 (D0), 4 (D0-D3), or 8 (D0-D7).

The basic operation on the multimedia card/SD/SD I/O bus is command/response structure.

The data transmitted on SD/SD I/O memory card is transmitted in the form of data block; the data transmitted on MMC is transmitted in the form of data block or data flow; the data transmitted on CE-ATA device is also transmitted in the form of data block.

Figure 120 SDIO "No-response" and "No-data" Operation

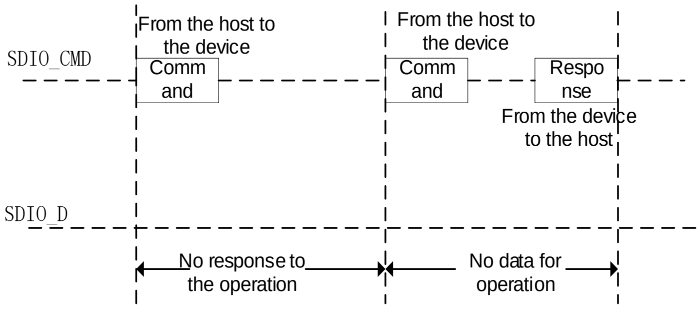

SDIO_CMD
From the host to the device
Command
From the host to the device
Command
Response
From the device to the host
SDIO_D
No response to the operation
No data for operation

Figure 121 SDIO "Multi-" Data Block Read Operation

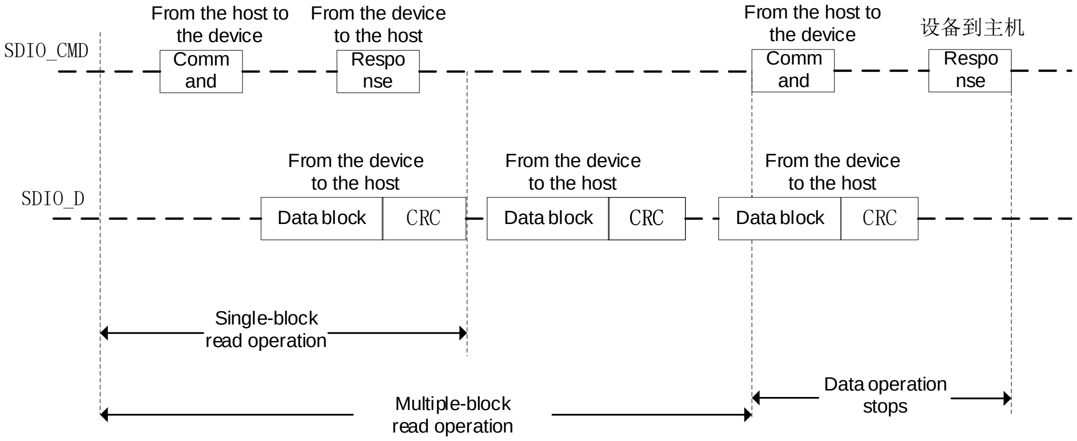
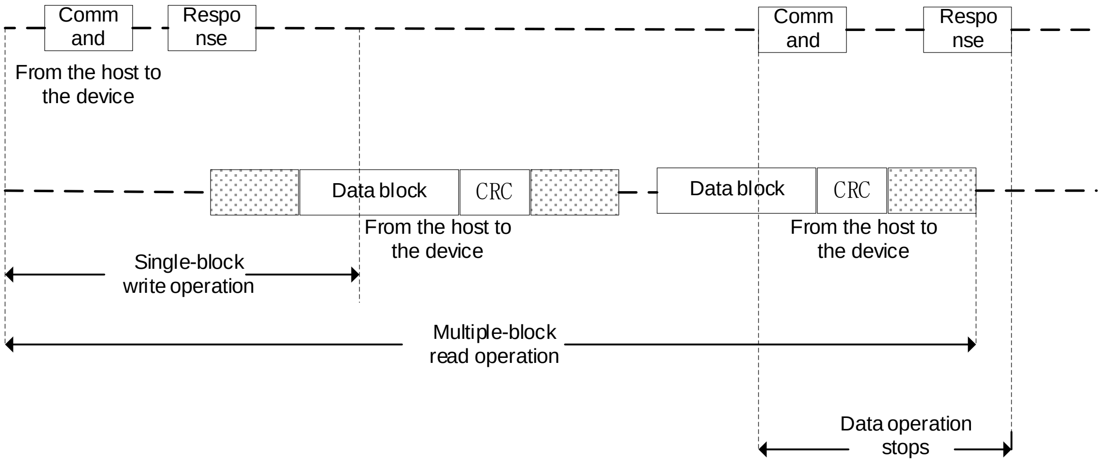

SDIO_CMD
From the host to the device
Command
From the device to the host
Response
From the host to the device
Command
设备到主机
Response
SDIO_D
From the device to the host
Data block
CRC
From the device to the host
Data block
CRC
From the device to the host
Data block
CRC
Single-block read operation
Multiple-block read operation
Data operation stops

Figure 122 SDIO "Multi-" Data Block Write Operation


Command
Response
Command
Response
From the host to the device
Data block
CRC
Data block
CRC
From the host to the device
From the host to the device
Single-block write operation
Multiple-block read operation
Data operation stops

▨: Busy

Figure 123 SDIO Data Flow Read Operation

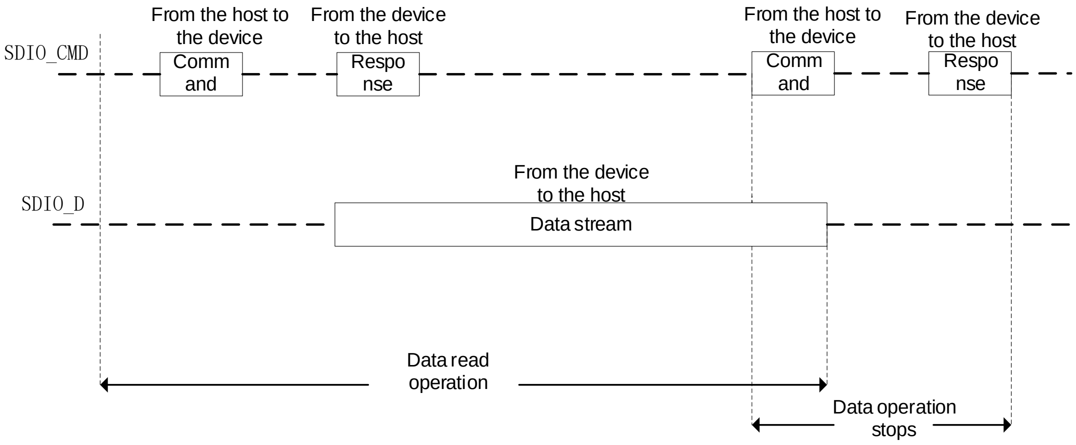
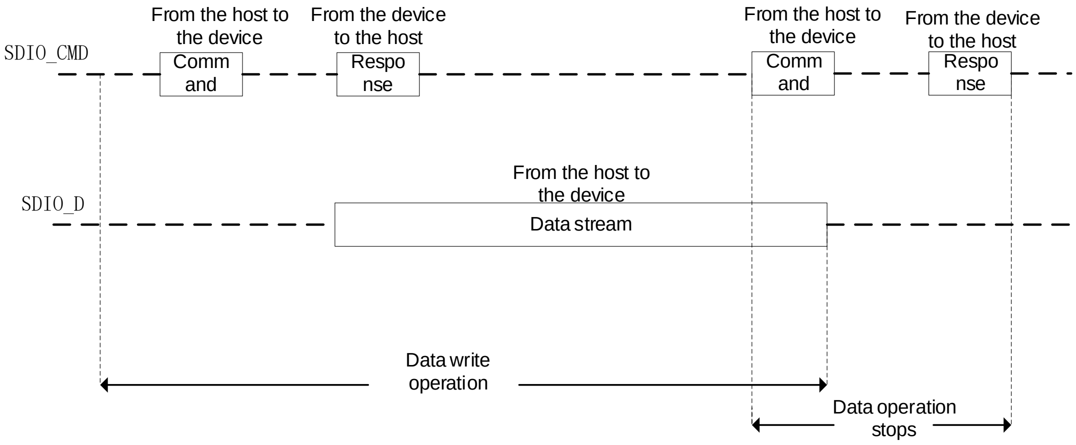

SDIO_CMD
From the host to the device
Command
From the device to the host
Response
From the host to the device
Command
From the device to the host
Response
SDIO_D
From the device to the host
Data stream
Data read operation
Data operation stops

Figure 124 SDIO Data Flow Write Operation


SDIO_CMD
From the host to the device
Command
From the device to the host
Response
From the host to the device
Command
From the device to the host
Response
SDIO_D
From the host to the device
Data stream
Data write operation
Data operation stops

25.4.2 SDIO Adapter

SDIO generates the following signals:

- SDIO_CLK: Clock provided by SDIO controller to the card. Each clock cycle directly sends 1-bit command or data on the command line (SDIO_ CMD) and all data lines (SDIO_ D). SDIO_ CLK frequency is within 0-20MHz for MMC card V3.31, within 0-48MHz for MMC card V4.2, and within 0-25MHz for SD or SD I/O card.
- SDIO_CMD: The signal is a bidirectional command channel, which is used for card initialization and command transmission. Command is transmitted from SDIO controller to the card, and the response is transmitted from the card to the master. There are two operation modes of CMD signal: the open-drain mode of initialization (used for MMC card V3.31 and previous versions) and the push-pull mode of command sending (SD, SD I/O, and MMC card V4.2 initialization are also push-pull mode).
- SDIO_D[7:0]: The signal lines are bidirectional data channels and in push-pull mode. By default, after power-on or reset, only D0 is used for data transmission. SDIO adapter can be configured with wider

data bus for data transmission, using D0-D3 and D0-D7 (only on MMC V4.2).

Control unit

The control unit includes power management and clock management functions.

Command unit

The command unit realizes sending commands to the card and receiving commands from the card, and the data transmission is controlled by the command state machine (CPSM).

Command state machine

CS_Idle Ready to send the command after reset

1.CPSM is enabled and WAIDEND is enabled → CS_Pend

2.CPSM is enabled and WAIDEND is disabled → CS_Send

3.CPSM is turned off → CS_Idle

Note: Command the state machine to maintain at least eight SDIO-CLK cycles when it is in idle state so as to meet the timing constraints of NCC and NRC

NCC：Minimum time interval between two host commands
NRC：Minimum time interval between host command and card response

CS_Pend
Wait for the end of data transmission

1.Data transmission is completed → CS_Send

2.CPSM is turned off → CS_Idle

CS_Send Send command

1.There is response after the command is sent → CS_Wait

2.There is no response after the command is sent → CS_Idle

3.CPSM is turned off → CS_Idle

CS_Wait Start bit of waiting for response

1.Response is received (the start bit is detected)	→ CS_Wait
2.Receiving response timeout	→ CS_Idle
3.CPSM is turned off	→ CS_Idle

Note: The command timeout is fixed to 64 SDIO-CLK clock cycles

CS_Receive Response is received and CRC is detected

1. The response is received in CE-ATA mode, CE-ATA interrupt is disabled, waiting for the command of CE-ATA device and the signal enable is completed	→CS_Waitcompl
2. The response is received in CE-ATA mode, CE-ATA interrupt is disabled, waiting for the command of CE-ATA device and the signal disable is completed	→ CS_Pend
3. CPSM is turned off	→CS_Idle
4. Response is received	→CS_Idle
5. Command CRC detection failed	→CS_Idle

CS_Waitcampl Wait for the signal for completion of the command of CE-ATA device

1.The signal for completion of the command of CE-ATA is received	→ CS_Idle
2.CPSM is turned off	→ CS_Idle
3.Command CRC detection failed	→ CS_Idle

Data unit

The data unit realizes the data transmission between the master and the card. When the data width is 8 bits, SDIO_D[7:0] signal line is used for data transmission. When the data width is 4 bits, SDIO_D[3:0] signal line is used for data transmission. When the data width is 1 bit, SDIO_D[0] signal line is used for data transmission.

Data transmission stream is controlled by data state machine (DPSM).

Data state machine

DS_Idle The data unit is idle, waiting to send and receive data

1. DPSM is enabled and data are transmitted from the host to the card	→ DS_WaitS
2. DPSM is enabled and data are transmitted from the card to the host	→ DS_WaitR
3. DPSM is enabled, read wait has started and SDIO mode is enabled	→ DS_Readwait

DS_WaitS
The flag that wait-for-data FIFO is empty is invalid or the data transmission ends

1. Data transmission ends → DS_Idle

2. DPSM is turned off → DS_Idle

3. The flag that the data FIFO is empty is invalid → DS_Send

DS_Send Send data to the card

1. Data have been sent → DS_Busy

2. DPSM is turned off → DS_Idle

3. Data FIFO underrun error is sent → DS_Idle

4. Internal CRC error → DS_Idle

DS_Busy State flag of waiting for CRC

1. Data have been sent → DS_Busy

2. DPSM is turned off → DS_Idle

3. Data FIFO underrun error is sent → DS_Idle

4. Internal CRC error → DS_Idle

DS_WaitR Start bit of waiting to receive data

1.Data receiving ends → DS_Idle

2.DPSM is turned off → DS_Idle

3.Data timeout → DS_Idle

4.The start bit is received before timeout → DS_Receive

DS_Receive

Receive data from the card and write them to the data FIFO

1. Data block has been received → DS_WaitR
2. Data transmission ends → DS_WaitR
3. Data FIFO overrun is sent → DS_Idle
4. Data have been received, read wait starts and SDI/o mode is enabled → DS_Readwait
5. DPSM is turned off or there is CRC error → DS_Idle

DS_Readwait Wait for "read wait stop" command

1. "Read wait stop" is enabled → DS_WaitR
2. DPSM is turned off → DS_Idle

25.4.3 AHB Interface

AHB interface can not only access to SDIO registers anddata FIFO, but also generate interrupts and DMA requests. It includes data FIFO unit, register unit and interrupt/DMA request control logic.

SDIO interrupt

When at least one of the selected state flags is high, the interrupt logic will generate an interrupt request. Interrupt enable register enables the interrupt logic to generate corresponding interrupt.

Data FIFO

The data FIFO unit has a data buffer area for receive and transmit FIFO. FIFO includes a data buffer with 32-bit width for each word and with depth of 32 words. The transmit FIFO is used to write data to the card; the data to be transmitted is written into the transmit FIFO through AHB bus; the data unit in SDIO adapter reads data from the transmit FIFO and sends the data to the card. The receive FIFO is used to read data from the card, and then write the data to be transmitted to the receive FIFO.

Register unit

The register unit includes all system registers and generates signals for communication between the control card and the controller.

25.4.4 Card Function Description

25.4.4.1 Card register

The card internally defines the interface registers: OCR, CID, CSD, EXT_CSD, RCA, DSR and SCR. These registers can access only through corresponding commands. OCR, CID, CSD and SCR registers include the specific information of the card. RCA and DSR registers are configuration registers for storing the actual configuration parameters. EXT_CSD register includes both the specific information of the card and the actual structure parameters.

OCR register: 32-bit operating condition register stores V_{DD} voltage description and memory mode indication (MMC) of the card. In addition, the register includes a state information bit. If the card power-on process has been completed, this state bit will be set. This register is slightly different between MMC and SD card. The master can use CMD1 (MMC), ACMD41 (SD memory card), and CMD5 (SD I/O) to capture the content of this register.

CID register: The card identification register (CID) is 128-bit wide. It contains the card identification information used in the card identification phase. Each read/write (RW) card shall have a unique identification number.

CSD register: The specific card data register provides the content information in the access card. CSD defines data format, error correction type, maximum data access time, and data transmission speed.

Extended CSD register: Only MMC4.2 has this register. Extended CSD register defines the card attributes and selection mode. Its length is 512 bytes. At most 320 bytes are the attribute section, which defines the function of the card and cannot be modified by the master. At least 192 bytes are the mode section, which defines under which configuration the card works.

RCA register: The writable 16-bit relative card address register stores the card address, which is released by the card during card initialization. This address is used for communication between the addressing master and the card after the card identification process.

DSR register: 16-bit drive stage register, which can be used to improve bus performance in extended operating conditions (depending on bus length, transmission rate and other parameters).

SCR register: Only SD/SD I/O has this register.

25.4.4.2 Command

The command for the control card consists of four different types:

Command type	Meaning
Broadcast command (BC)	Send to all cards, but there is no response returned

Command type	Meaning
Broadcast command with response (BCR)	Send to all cards and receive response from all cards
Addressing (point-to-point) command (AC)	Send to addressing card, and there is no data transmission on SDIO_D line
Addressing (point-to-point) data transmission command (ADTC)	Send to addressing card, and there is data transmission on SDIO_D line

Command format

All command formats are 48-bit fixed code length, which requires 1.92us (25MHz), 0.96us (50MHz) and 0.92us (52MHz) transmission time.

Table 110 Command Format

Bit	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Numerical Value	0	1	-	-	-	1
Instruction	Start Bit	Transmission Bit	Command Index	Parameter	CRC7	End bit

SD I/O supports two types of response, both of which support CRC error detection.

- 48-bit short response
- 136-bit long response

Table 111 Short Response Format

Bit	47	46	[45:40]	[39:8]	[7:1]	0
Width	1	1	6	32	7	1
Numerical Value	0	0	-	-	-	1
Instruction	Start Bit	Transmission Bit	Command Index	Parameter	CRC7 or (1111111)	End bit

Table 112 Long Response Format

Bit	135	134	[133:128]	[127:1]	0
Width	1	1	6	127	1
Numerical Value	0	0	111111	-	1
Instruction	Start Bit	Transmission Bit	Reserved	CID or CSD	End bit

Command description

Table 113 Basic Command

Command Index	Type	Parameter	Response format	Short name	Instruction
CMD0	bc	[31:0] Stuffing bit	-	GO_IDLE_STATE	Reset all cards to the idle state.
CMD1	bc	[31:0]OCR	R3	SEND_OP_COND	In the idle state, request the card to send the response (including the content of the operating condition register) through the CMD line
CMD2	bcr	[31:0] Stuffing bit	R2	ALL_SEND_CID	Request any card to send CID data through the CMD line (all cards connected to the master will respond)
CMD3	bcr	[31:0] Stuffing bit	R6	SEND_RELATIVE_ADDR	Request the card to release new relative card address (RCA)
CMD4	bc	[31:16]DSR [15:0]Stuffing bit	-	SET_DSR	Set DSR registers of all cards.
CMD5	bcr	[31:25]Reserved bit [24]S18R [23:0]I/O OCR	R4	IO_SEND_OP_COND	Only apply to I/O card. Query voltage range of all IO cards.
CMD6	ac	[31:26]Set to 0 [25:24]Access [23:16]Index [15:8]Value [7:3]Set to 0 [2:0]Command set	R1b	SWITCH	Only apply to MMC card. Switch the operation mode of the selected card or modify EXT_CSD register.
CMD7	ac	[31:16]RCA [15:0]Stuffing bit	R1b	SELECT/DESELECT_CARD	Used for switching of the card state.

Command Index	Type	Parameter	Response format	Short name	Instruction
CMD8	bcr	[31:12]Reserved bit [11:8]Operating voltage (VHS) [7:0]Inspection mode	R7	SEND_IF_COND	Send interface conditions to SD card, including master power supply voltage information and query whether the card supports the voltage. The reserved bit should be set to 0.
CMD9	ac	[31:16]RCA [15:0]Stuffing bit	R2	SEND_CSD	The selected card sends to its card specific data (CSD) through CMD
CMD10	ac	[31:16]RCA [15:0]Stuffing bit	R2	SEND_CID	The selected card sends to its card identification (CID) through CMD
CMD12	ac	[31:0] Stuffing bit	R1b	STOP_TRANSMISSION	Force the card to stop transmission.
CMD13	ac	[31:16]RCA [15:0]Stuffing bit	R1	SEND_STATUS	The selected card sends its state register.
CMD14	adtc	[31:0] Stuffing bit	R1	BUSTEST_R	The master reads the reverse bus test data mode from the card.
CMD15	ac	[31:16]RCA [15:0]Stuffing bit	-	GO_INACTIVE_STATE	Convert the selected card to inactive state.
CMD19	adtc	[31:0] Stuffing bit	R1	BUSTEST_W	The master sends the bus test mode to the card.

Table 114 Block-oriented Write Command

Command Index	Type	Parameter	Response format	Short name	Instruction
CMD23	ac	[31:16]Set to 0 [15:0]Number of blocks	R1	SET_BLOCK_COUNT	Define the number of blocks to be transferred in subsequent blocks or write commands.
CMD24	adtc	[31:0]Data address	R1	WRITE_BLOCK	Write a block according to the selected length of SET_BLOCKLEN command.
CMD25	adtc	[31:0]Data address	R1	WRITE_MULTIPLE_BLOCK	Continue to write the data block until receiving STOP_TRANSMISSION command or achieving the specified number of blocks.
CMD26	adtc	[31:0] Stuffing bit	R1	PROGAM_CID	Program the card identification register. This command can be transmitted to each card once only. The programming involves hardware changes to prevent subsequent operations after the first programming. Reserved to the manufacturer through this command.
CMD27	adtc	[31:0] Stuffing bit	R1	PROGAM_CSD	Program the programmable bits in the card CSD.

Command Index	Type	Parameter	Response format	Short name	Instruction
CMD28	ac	[31:0]Data address	R1b	SET_WRITE_PROT	If the card has write protection function, this command will set the write protection bit of the specified group. The write protection characteristic is set in the special data area of the card (WP_GRP_SIZE)
CMD29	ac	[31:0]Data address	R1b	CLR_WRITE_PROT	If the card has write protection function, this command will clear the write protection bit of the addressing group.
CMD30	adtc	[31:0]Write protection data address	R1	SEND_WRITE_PROT	If the card has write protection function, the command requests the card to send write protection bit state.

25.4.5 Specific Operations

25.4.5.1 SD I/O card operations

SD I/O card (including IO card and combined card) supports the following specific operations:

- Read wait operation
- Pause/Recovery operation
- Interrupt

SDIO supports these operations only when setting SDIOEN bit in SDIO_DCTRL; except for read pause, it does not require special hardware operation.

SD I/O read wait operation

Read wait (RW) is only used for SD I/O 1-bit and 4-bit modes. Read wait operation allows a master to send a signal to stop data transmission when performing the operation of reading multiple blocks for the card, and allows the master to send commands to any function in the SD I/O card.

Before receiving the first data block, it is allowed to start the read wait process, enable the data channel (DTEN bit is set to 1), enable SDIO specific operation (SDIOEN bit is set to 1), and start read wait. At the same time, the data are transmitted from the card to the SDIO master, and DPSM will directly enter the read wait state from idle state.

In the read wait state, after 2 SDIO_CK clock cycles, DPSM drive SDIO_ D2 is 0. In this state, if RWSTOP bit is set, DPSM will stay for 2 more SDIO_CK clock cycles in the waiting state, and (according to SDIO specification) the drive SDIO_D2 is 1 in one clock cycle. Then DPSM starts to wait to receive data from the card. When receiving data block, DPSM will not enter the read wait state even if the start wait is set.

SDIO card cannot perform the above read wait operation, SDIO can stop SDIO_CK entering the read wait and two SDIO_CK cycles later after receiving the current data block, DPSM stops the clock and recovers the clock after setting the read wait start bit.

Figure 125 Read Wait Operation Using SDIO_D2 Signal Line

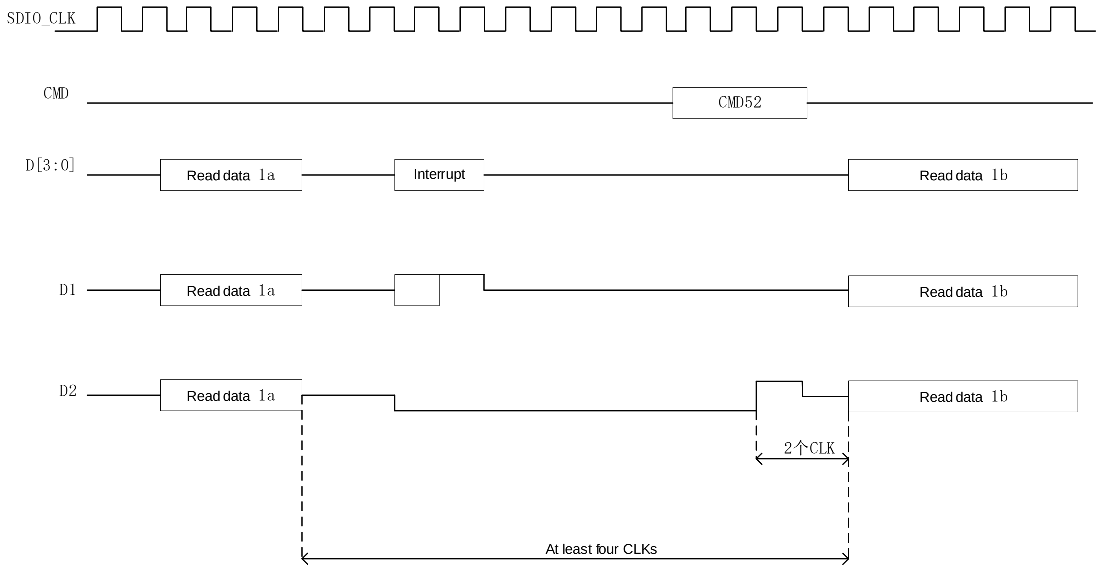
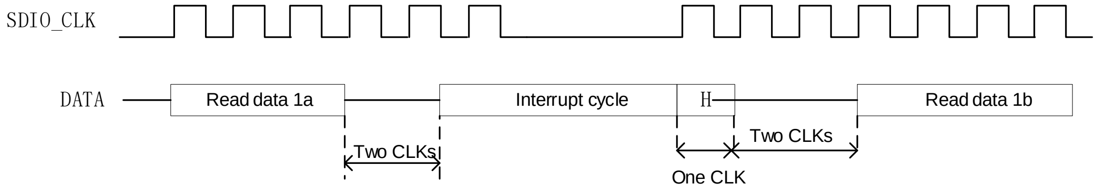


Figure 126 Read Wait Operation through by Stopping SDIO_CLK



SDIO interrupt

When the SDIOEN bit is set, the SDIO master detects SDIO interrupt on SDIO_D signal line.

25.4.5.2 CE-ATA specific operation

CE-ATA device supports the following specific operation:

- Receive command completion signa
- Send command completion and closing signal

SDIO supports these operations only when SDIO_CMD ATACMD bit is set.

Command completion signal

CE-ATA defines the command completion signal. The device uses this signal to notify the master that the ATA command is completed or encounters an error, and ATA command is terminated

Command completion and closing signal

The master can send a command to complete the function of closing the signal and canceling the device return command to complete the signal.

If the "Enable CMD end bit" of ATACMD is not set and the "Non-terminal enable bit" of INTDIS is set, the command completion and closing signal will be transmitted 8-bit cycle later after receiving a short response.

25.4.6 Card State Register Functional Description

The response format R1 contains a 32-bit card state field, which is used to send the card state information to the card master (the information may be stored in the local state register). Unless otherwise specified, the state returned by the card is always related to the previous command.

The following table defines different state information:

Table 115 Different State Information Types

Abbreviation	Definition
E	Error bit
S	State bit
R	Detection bit, set according to the actual command response
X	Detection bit, set during the execution of command. The master of SDIO card reads these bits and queries the card state by sending the state command.

Table 116 Clear Conditions

Abbreviation	Definition
A	According to current state of the card
B	Always related to previous command. Can be cleared once correct command is received (with delay of one command).
C	Clear once read

Table 117 Card State

Bit	Name	Type	Numerical Value	Instruction	Clear conditions
1:0	Reserved for the test mode of the manufacturer.				
2	Reserved for the command related to application.				
3	AKE_SEQ_ERROR	ER	0=No error 1=Error	Verification sequence error.	C
4	Reserved for SD I/O card				
5	APP_CMD	SR	0=Not allowed 1=Allowed	The card expects ACMD, or the instruction command has been interpreted as an ACMD command.	C
6	Reserved				
7	SWITCH_ERROR	EX	0=No error 1=Conversion error	The card is not switched to the desired mode according to the requirements of SWITCH command.	B
8	READY_FOR_DATA	SR	0=Not ready 1=Ready	Corresponding to the empty signal of the buffer on the bus.	
12:9	CURRENT_STATE	SR	0=Idle 1=Ready 2=Identify 3=Standby 4=Send 5=Data 6=Receive 7=Program 8=Disconnect 9=Busy test 10~15=Reserved	The state of the state machine in the card when receiving a command. If the execution of a command causes a state change, the change will be reflected in the response of next command. These four bits are interpreted by decimal numbers 0 to 15.	B
13	ERASE_RESET		0=Clear 1=Set	Because a command other than the erase sequence (rather than CMD35, CMD36, CMD38 or CMD13) is received, the sequence of entering the erase process is aborted.	C
14	CARD_ECC_DISABLED	SX	0=Allowed 1=Not allowed	Internal ECC is not used during execution of the command.	A
15	WP_ERASE_SKIP	EX	0=Not protected 1=Protected	Encounter existing write protection data block, and only part of address space is erased.	C

Bit	Name	Type	Numerical Value	Instruction	Clear conditions
16	CID/CSD_OVERWRITE	EX	0=No error 1=Error	It can be any of the following errors: It has been written into the CID register and cannot be overridden The read-only part of the CSD does not match the content of the card Attempt to reverse copy or permanent write protection, i.e., restore or release the write protection.	C
17	Reserved				
18	Reserved				
19	ERROR	EX	0=No error 1=Error	Generate an error related to the previous master command executed (not defined in the standard) Error inside the card (such as read or write error).	C
20	CC_ERROR	ER	0=No error 1=Error	Error inside the card (not defined in the standard), which has nothing to do with the command of the master.	C
21	CARD_ECC_FAILED	EX	0=Succeeded 1=Failed	ECC check is implemented in the card, but it fails when correcting the data.	C
22	ILLEGAL_COMMAND	ER	0=No error 1=Error	The command is illegal for current card state.	B
23	COM_CRC_ERROR	ER	0=No error 1=Error	CRC parity error in previous command.	B
24	LOCK_UNLOCK_FAILED	EX	0=No error 1=Error	Sequence error of command or wrong password detected in lock/unlock.	C
25	CARD_IS_LOCKED	SR	0=Card unlocked 1=Card locked	When this bit is set, it means the card has been locked.	A
26	WP_VIOLATION	EX	0=No error 1=Error	Attempt to program the data block of a write protection.	C
27	ERASE_PARAM	EX	0=No error 1=Error	Illegal erase group selected when erasing.	C

Bit	Name	Type	Numerical Value	Instruction	Clear conditions
28	ERASE_SEQ_ERROR		0=No error 1=Error	The sequence of sending erase command is wrong.	C
29	BLOCK_LEN_ERROR		0=No error 1=Error	The parameters of SET_BLOCKLEN command exceed the maximum allowable range of the card, or the previously defined data block length is illegal for the current command (for example, the master sends a write command, and the current block length is less than the minimum allowable length of the card, while some data blocks are not allowed to be written).	C
30	ADDRESS_MISALIGN		0=No error 1=Error	The first block defined by the address parameter in the command (compared with the current block length) is not aligned with the physical block of the card. A multi-data block or data stream read/write operation (even starting from a legal address) attempts to read or write a data block that is not aligned with the physical block.	C
31	ADDRESS_OUT_OF_RANGE	ERX	0=No error 1=Error	The address parameter in the command is beyond the allowed range of the card. A multi-data block or data stream read/write operation (even starting from a legal address) attempts to read or write the part beyond the capacity of the card.	C

25.4.6.1 SD state register functional description

The SD state includes the state bits related to the specific functions of the SD memory card and some state bits related to future applications. The length of the SD state is a 512-bit data block. After receiving the ACMD13 command

(CMD55, then CMD13), the content of this register will be transmitted to the SDIO card master. The ACMD13 command can be transmitted only when the card is in transmission state (the card has been selected).

The following table defines different SD state register information.

Table 118 SD State Register Information

Abbreviation	Definition
E	Error bit
S	State bit
R	Detection bit, set according to the actual command response
X	Detection bit, set during the execution of command. The master of SDIO card reads these bits and queries the card state by sending the state command.

Note: For the abbreviations and definitions of related types and clear condition fields in the table, see.

Table 119 Clear Conditions

Abbreviation	Definition
A	According to current state of the card
B	Always related to previous command. Can be cleared once correct command is received (with delay of one command).
C	Clear once read

Table 120 SD State

Bit	Name	Type	Numerical Value	Instruction	Clear conditions
311:0	Reserved to the manufacturer				
399:312	Reserved				
401:400	ERASE_OFFSET				
407:402	ERASE_TIMEOUT	S R	Fixed offset value added in erase	(See the instructions below)	A
423:408	ERASE_SIZE	S R	Erase the timeout value of specified range of UNIT_OF_ERASE_AU	(See the instructions below)	A
427:424	Reserved	S R	The number of AU that can be erased at a time	(See the instructions below)	A
431:428	AU_SIZE	S R	AU size (see the instructions below)	(See the instructions below)	A

Bit	Name	Type	Numerical Value	Instruction	Clear conditions
439:432	PERFORMANCE_MOVE	S R	Transmission performance in 1MB/s (see the instructions below)	(See the instructions below)	A
447:440	SPEED_CLASS	S R	Speed type of the card (see the instructions below)	(See the instructions below)	A
479:448	SIZE_OF_PROTECTED_AREA	S R	Size of protected area (see the instructions below)	(See the instructions below)	A
495:480	SD_CARD_TYPE	S R	'00xxh'= SD memory card in physical specification version 1.01~2.00 ('x' means any value). The defined cards are: '0000'= Universal SD read-write card '0001'= SD ROM card	The low 8 bits of this field can define different variants of SD memory card in the future (each bit can be used to define different SD types). The high 8 bits can be used to define SD cards that do not comply with the current SD physical layer specification.	A
508:496	Reserved				
509	SECURED_MODE	S R	0=Not in the secret mode 1=In the secret mode	The card is in secret operation mode (see "SD Confidentiality Specification").	A
511:510	DAT_BUS_WIDTH	S R	00=1 (default) 01-Reserve 10=4-bit width 11=Reserved	Current data bus width defined by SET_BUS_WIDTH command.	A

SIZE_OF_PROTECTED_AREA

The way of setting this bit is different between standard-capacity card and high-capacity card:

- For the standard-capacity card, the capacity of the protected area is calculated as follows:
  Protected area=SIZE_OF_PROTECTED_AREA * MULT * BLOCK_LEN
  The unit of SIZE_OF_PROTECTED_AREA is MULT * BLOCK_LEN.
- For the high-capacity card, the capacity of the protected area is calculated as follows:
  Protected area=SIZE_OF_PROTECTED_AREA
  The unit of SIZE_OF_PROTECTED_AREA is byte.

SPEED_CLASS

These 8 bits indicate the type of speed and can be used for calculating the value of PW/2 (PW is the performance of write).

Table 121 Speed Type Code

SPEED_CLASS	Definition of Numerical Value
00h	Type 0
01h	Type 2
02h	Type 4
03h	Type 6
04h~FFh	Reserved

PERFORMANCE_MOVE

These 8 bits indicate mobility performance (Pm) in 1MB/s. If the card does not use RU (record unit) mobile data, it should be considered that Pm is infinitely great. Setting this field for FFH means infinitely great.

Table 122 Mobility Perfromance Code

PERFORMANCE_MOVE	Definition of Numerical Value
00h	Undefined
01h	1MB/s
02h	2MB/s
……	……
Feh	254MB/s
FFh	Infinitely great

AU_SIZE

These four bits indicate the length of AU, and the value is the multiple of the power of 2 in 16KB.

Table 123 AU_SIZE Codes

AU_SIZE	Definition of Numerical Value
00h	Undefined
01h	16KB
02h	32KB
03h	64KB
04h	128KB
05h	256KB
06h	512KB
07h	1MB
08h	2MB
09h	4MB
Ah~Fh	Reserved

According to the capacity of the card, the maximum AU length is defined by the following table. The card can set any AU length between the RU length and the maximum AU length.

Table 124 Maximum AU Length

Capacity	Maximum AU length
16MB~64MB	512KB
128MB~256MB	1MB
512MB	2MB
1GB~32GB	4MB

ERASE_SIZE

This 16-bit field gives the value of N_{ERASE}, and when N_{ERASE} number of AU is erased, ERASE_TIMEOUT defines the timeout period.

If the master can determine the value of N_{ERASE} in a certain erase, the erase progress can be displayed.

Table 125 ERASE_SIZE Codes

ERASE_SIZE	Definition of Numerical Value
0000h	Overtime calculation not supporting erase
0001h	One AU
0002h	Two AUs
0003h	Three AUs
……	……
FFFFh	65535 AUs

ERASE_TIMEOUT

These 6 bits give T_{ERASE}. When multiple AUs are erased, T_{ERASE} gives the erase timeout calculated from offset.

Table 126 Erase Timeout Codes

ERASE_TIMEOUT	Definition of Numerical Value
00	Overtime calculation not supporting erase
01	1 second
02	2 seconds
03	3 seconds
……	……
63	63 seconds

ERASE_OFFSET

These two bits give TOFFSET, and when both ERASE_SIZE and ERASE_TIMEOUT are 0, this value is meaningless.

Table 127 Erase Offset Codes

ERASE_OFFSET	Definition of Numerical Value
0	0 second
1	1 second
2	2 seconds
3	3 seconds

25.5 Register Address Mapping

Table 128 SDIO Register Address Mapping

Register name	Description	Offset address
SDIO_PWRCTRL	SDIO power control register	0x00
SDIO_CLKCTRL	SDIO clock control register	0x04
SDIO_ARG	SDIO parameter register	0x08
SDIO_CMD	SDIO command register	0x0C
SDIO_CMDRES	SDIO command response register	0x10
SDIO_RESx	SDIO response register x	0x14 + 4* (x-1), wherein x=1 to 4
SDIO_DATATIME	SDIO data timer register	0x24
SDIO_DATALEN	SDIO data length register	0x28

Register name	Description	Offset address
SDIO_DCTRL	SDIO data control register	0x2C
SDIO_DCNT	SDIO data counter register	0x30
SDIO_STS	SDIO state register	0x34
SDIO_ICF	SDIO clear interrupt register	0x38
SDIO_MASK	SDIO interrupt mask register	0x3C
SDIO_FIFOCNT	SDIO counter register	0x48
SDIO_FIFODATA	SDIO data FIFO register	0x80

25.6 Register Functional Description

The device communicates with the system through the 32-bit control registers that can be operated on AHB. These peripheral registers must be operated in word (32-bit) mode.

25.6.1 SDIO power control register (SDIO_PWRCTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
1:0	PWRCTRL	R/W	Power Supply Control Select current function state of card clock. 00: Power off, card cock stopped. 01: Reserved. 10: Reserved power-on state. 11: Power-on state, card clock started.
31:2	Reserved		

Note: This register cannot be written within 7 HCLK clock cycles after data write.

25.6.2 SDIO clock control register (SDIO_CLKCTRL)

Offset address: 0x04

Reset value: 0x0000 0000

SDIO_CLKCTRL register controls SDIO_CLK to output the clock.

Field	Name	R/W	Description
7:0	CLKDIV	R/W	Clock Divide Factor This domain defines the division factor between input clock (SDIOCLK) and output clock (SDIO_CLK): SDIO_CLK frequency=SDIOCLK/[CLKDIV + 2].
8	CLKEN	R/W	Clock Enable 0: Disable 1: Enable

Field	Name	R/W	Description
9	PWRSAV	R/W	Power Saving Mode Configuration Reduce power consumption by disabling SDIO_ CLK outputs beyond the bus activity. 0: Enable 1: Disable
10	BYPASSEN	R/W	Clock Divider Bypass Enable Before driving SDIO_CLK to output signals, it is required to divide the frequency of SDIOCLK; however, if the divider is bypassed, SDIOCLK will directly drive SDIO_CLK to output signals. 0: Disable 1: Enable
12:11	WBSEL	R/W	Wide Bus Mode Select Select bus mode for different bits, corresponding to different SDIO_D bits. 00: Default, using SDIO_D0 01: 4 bits, using SDIO_D[3:0] 10: 8 bits, using SDIO_D[7:0] 11: Reserved
13	DEPSEL	R/W	SDIO_CLK Dephasing Select Select SDIOCLK rising edge or falling edge to generate SDIO_CLK. 0: Rising edge 1: Falling edge
14	HFCEN	R/W	HW Flow Control Enable 0: Disable 1: Enable
31:15	Reserved		

Note:

(1) When SD/SD I/O card or multimedia card is in identification mode, the frequency of SDIO_CLK must be less than 400kHz.

(2) When all cards have been assigned the corresponding address, the clock frequency can be changed to the maximum frequency allowed by the card bus.

(3) This register cannot be written within 7 HCLK clock cycles after writing data. For SD I/O card, SDIO_CLK can be stopped during read wait period, and then SDIO_CLKCTRL register does not control SDIO_CLK.

25.6.3 SDIO parameter register (SDIO_ARG)

Offset address: 0x08

Reset value: 0x0000 0000

Command parameters are also part of the command, and SDIO_ARG register contains 32-bit command parameters and is transmitted to the card together with the command.

Field	Name	R/W	Description
31:0	CMDARG	R/W	Command Argument Store the command parameters.

25.6.4 SDIO command register (SDIO_CMD)

Offset address: 0x0C

Reset value: 0x0000 0000

SDIO_CMD register contains command index and command type bit.

Field	Name	R/W	Description
5:0	CMDINDEX	R/W	Command Index The command index, as part of the command, is send to the card together with the command.
7:6	WAITRES	R/W	Wait for Response Indicate whether CPSM needs to wait for the response, and if it needs to wait for the response, the response type will be indicated. 00: No response, expecting CMDSENT flag 01: Short response, expecting CMDREND or CCRCFAIL flag 10: No response, expecting CMDSENT flag 11: Long response, expecting CMDREND or CCRCFAIL fag
8	WAITINT	R/W	CPSM Waits for Interrupt Request CPSM enables or disables the command timeout control and waits for interrupt request. 0: Enable 1: Disable
9	WENDDATA	R/W	CPSM Waits for Ends of Data Transfer (CmdPend Internal Signal) 0: Invalid 1: CPSM waits for the end of data transmission before starting to send a command.
10	CPSMEN	R/W	Command Path State Machine (CPSM) Enable Enable CPSM. 0: Disable 1: Enable
11	SDIOSC	R/W	SD I/O Suspend Command 0: Invalid 1: The command to be transmitted is a suspend command (only used for SD I/O card).
12	CMDCPEN	R/W	Enable CMD Completion Enable command completion signal. 0: Disable 1: Enable
13	INTEN	R/W	Interrupt Enable 0: Enabl 1: Disable
14	ATACMD	R/W	CE-ATA Command 0: Invalid 1: Switch CPSM to CMD61:

Field	Name	R/W	Description
31:15	Reserved		

Note:

(1) This register cannot be written within 7 HCLK clock cycles after writing data.

(2) The multimedia card can send two kinds of responses: 48-bit short response or 136-bit long response. The SD card and SD I/O card can only send short response, parameters can be changed according to the type of response, and the software will distinguish the type of response according to the command transmitted. CE-ATA devices only send short responses.

25.6.5 SDIO command response register (SDIO_CMDRES)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
5:0	CMDRES	R	Response Command Index Store the command index in the finally received command response.
31:6	Reserved		

25.6.6 SDIO response register x (SDIO_RESx)

Offset address: 0x14 + 4* (x-1), wherein x=1…4

Reset value: 0x0000 0000

SDIO_RES1/2/3/4 register contains the card state, namely, some information about the received response.

Field	Name	R/W	Description
31:0	CARDSTSx	R	See the table below.

The state length of the card is 32 bits or 127 bit according to the response state.

Table 129 Response Type and SDIO_RESx Register

Register	Short response	Long response
SDIO_RES1	Card state [31:0]	Card state [127:96]
SDIO_RES2	Unused	Card state [95:64]
SDIO_RES3	Unused	Card state [63:32]
SDIO_RES4	Unused	Card state [31:1]

Always receive the most significant bit of card state first, and the lowest bit of SDIO_RES3 register is always 0.

25.6.7 SDIO data timer register (SDIO_DATATIME)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	DATATIME	R/W	Data Timeout Period Record the data timeout period in card bus clock cycle.

Note: Before writing the data control register for data transmission, first write the data timer register and data length register.

25.6.8 SDIO data length register (SDIO_DATALEN)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
24:0	DATALEN	R/W	Data Length Byte length of data to be transmitted.
31:25	Reserved		

Note: For block data transfer, the value in SDIO_DATALEN must be a multiple of the data block length. Before writing SDIO_DCTRL for data transmission, first write SDIO_DATATIME and SDIO_DATALEN.

25.6.9 SDIO data control register (SDIO_DCTRL)

Offset address: 0x2C

Reset value: 0x0000 0000

SDIO_DCTRL register control data channel state channel (DPSM).

Field	Name	R/W	Description
0	DTEN	R/W	Data Transfer Enabled 0: Disable 1: Enable
1	DTDRCFG	R/W	Data Transfer Direction Configuration 0: From controller to card 1: From card to controller
2	DTSEL	R/W	Data Transfer Mode Select 0: Block data transfer 1: Stream data transfer
3	DMAEN	R/W	DMA Enable 0: Disable 1: Enable
7:4	DBSIZE	R/W	Data Block Size Define the data bock length: 0000: Block length=20=1 byte 0001: Block length=21=2 bytes 0010: Block length=22=4 bytes 0011: Block length=23=8 bytes 0100: Block length=24=16 bytes 0101: Block length=25=32 bytes 0110: Block length=26=64 bytes 0111: Block length=27=128 bytes 1000: Block length=28=256 bytes 1001: Block length=29=512 bytes

Field	Name	R/W	Description
			1010: Block length=210=1024 bytes 1011: Block length=211=2048 bytes 1100: Block length=212=4096 bytes 1101: Block length=213=8192 bytes 1110: Block length=214=16384 bytes 1111: Reserved
8	RWSTR	R/W	Read Wait Start 0: Invalid 1: Start read wait operation
9	RWSTOP	R/W	Read Wait Stop Enable If RWSTR is set, stop read operation can be enabled. 0: Disable 1: Enable
10	RDWAIT	R/W	Read Wait Mode 0: Control stop SDIO_D2 1: Control use SDIO_CK
11	SDIOF	R/W	SD I/O Enable Functions If this bit is set, DPSM wil execute specific operation of SD I/O card.
31:12	Reserved		

Note: This register cannot be written within 7 HCLK clock cycles after writing data.

25.6.10 SDIO data counter register (SDIO_DCNT)

Offset address: 0x30

Reset value: 0x0000 0000

When DPSM enters Wait_R or Wait_S state from the idle state, SDIO_DCNT loads values from SDIO_DATALEN, and in the data transmission process, the counter will count down to 0, and then DPSM will enter the idle state and set DATAEND flag.

Field	Name	R/W	Description
24:0	DATACNT	R	Data Count Number Store the number of data bytes to be transmitted.
31:25	Reserved		

Note: This register can be read only by the end of data transmission.

25.6.11 SDIO state register (SDIO_STS)

Offset address: 0x34

Reset value: 0x0000 0000

SDIO_STS is a read-only register, and it contains two kinds of flags:

(1) Static flag (bit [23:22, 10:0]): Write SDIO interrupt clear register to clear these bits.

(2) Dynamtic flag (bits [21:11]: The state of these bits changes according to the logic of corresponding part.

Field	Name	R/W	Description
0	COMRESP	R	Command Response Received (CRC detection failure)
1	DBDR	R	Data Block Sent/Received (CRC detection failure)
2	CMDRESTO	R	Command Response Timeout Command timeout period is 64 SDIO_CLK clock cycles.
3	DATATO	R	Data Timeout
4	TXUDRER	R	Transmit FIFO Underrun Error
5	RXOVRER	R	Received FIFO Overrun Error
6	CMDRES	R	Command Response (CRC detection succeeded)
7	CMDSENT	R	Command Sent (No Response Required)
8	DATAEND	R	Data end (data counter, SDIO_DCNT=0)
9	SBE	R	Start Bit Not Detected On All Data Signals In Wide Bus Mode
10	DBCP	R	Data Block Sent/Received
11	CMDACT	R	Command Transfer In Progress
12	TXACT	R	Data Transmit In Progress
13	RXACT	R	Data Receive In Progress
14	TXFHF	R	Transmit FIFO Half Empty) At least more 8 words can be written in FIFO.
15	RXFHF	R	Receive FIFO Half Full There are at least eight words in FIFO.
16	TXFF	R	Transmit FIFO Full
17	RXFF	R	Receive FIFO Full If the hardware flow control is used, the RXFF signal will become effective when the FIFO is still 2 words short.
18	TXFE	R	Transmit FIFO Empty If the hardware flow control is used, the TXFE signal will become effective when the FIFO contains 2 words.
19	RXFE	R	Receive FIFO Empty
20	TXDA	R	Data Available In Transmit FIFO
21	RXDA	R	Data Available In Receive FIFO
22	SDIOINT	R	SDIO Interrupt Received
23	ATAEND	R	CE-ATA Command Completion Signal Received For CMD61
31:24	Reserved		

25.6.12 SDIO clear interrupt register (SDIO_ICF)

Offset address: 0x38

Reset value: 0x0000 0000

SDIO_ICF is a write-only register, and the corresponding bit in SDIO_STS state register will be cleared in corresponding register bit.

Field	Name	R/W	Description
0	DBCE	R/W	DBCE Flag Clear Clear DBCE flag. 0: Invalid 1: Clear
1	CRCE	R/W	CRCE Flag Clear Clear CRCE flag. 0: Invalid 1: Clear
2	CRTO	R/W	CRTO Flag Clear Clear CRTO flag. 0: Invalid 1: Clear
3	DTO	R/W	DTO Flag Clear Clear DTO flag. 0: Invalid 1: Clear
4	TXFUE	R/W	TXFUE Flag Clear Clear TXFUE flag. 0: Invalid 1: Clear
5	RXFOE	R/W	RXFOE Flag Clear Clear RXFOE flag. 0: Invalid 1: Clear
6	CMDRES	R/W	CMDRES Flag Clear Clear CMDRES flag. 0: Invalid 1: Clear
7	CMDSENT	R/W	CMDSENT Flag Clear Clear CMDSENT flag. 0: Invalid 1: Clear
8	DATAEND	R/W	DATAEND Flag clear Clear DATAEND flag. 0: Invalid 1: Clear
9	SBE	R/W	SBE Flag Clear Clear SBE flag. 0: Invalid 1: Clear
10	DBCP	R/W	DBCP Flag Clear Clear DBCP flag. 0: Invalid 1: Clear
21:11	Reserved		

Field	Name	R/W	Description
22	SDIOIT	R/W	SDIOIT flag clear bit Clear SDIOIT flag. 0: Invalid 1: Clear
23	ATAEND	R/W	ATAEND flag clear bit Clear ATAEND flag. 0: Invalid 1: Clear
31:24	Reserved		

25.6.13 SDIO interrupt mask register (SDIO_MASK)

Offset address: 0x3C

Reset value: 0x0000 0000

Set or cleared by software.

When the corresponding bit is set to 1, SDIO_MASK interrupt mask register decides which state bit generates an interrupt.

Field	Name	R/W	Description
0	CCRCFAIL	R/W	Command CRC Fail Interrupt Enable Enable/Disable command block CRC detection failure interrupt. 0: Disable 1: Enable
1	DCRCFAIL	R/W	Data CRC Fail Interrupt Enable Enable/Disable data block CRC detection failure interrupt. 0: Disable 1: Enable
2	CMDTO	R/W	Command Timeout Interrupt Enable Enable/Disable command timeout interrupt. 0: Disable 1: Enable
3	DATATO	R/W	Data timeout interrupt enable) Enable/Disable data timeout interrupt. 0: Disable 1: Enable
4	TXURER	R/W	Tx FIFO Underrun Error Interrupt Enable Enable/Disable transmit FIFO underrun error interrupt. 0: Disable 1: Enable
5	RXORER	R/W	Rx FIFO Overrun Error Interrupt Enable Enable/Disable receive FIFO overrun error interrupt. 0: Disable 1: Enable
6	CMDRESRC	R/W	Command Response Received Interrupt Enable Enable/Disable receiving response interrupt. 0: Disable 1: Enable

Field	Name	R/W	Description
7	CMDSENT	R/W	Command Sent Interrupt Enable Enable/Disable command transmitted interrupt since the software has set/cleared this bit. 0: Disable 1: Enable
8	DATAEND	R/W	Data End Interrupt Enable Enable/Disable data transmission end interrupt. 0: Disable 1: Enable
9	STRTER	R/W	Start Bit Error Interrupt Enable Enable/Disable start bit error interrupt. 0: Disable 1: Enable
10	DBEND	R/W	Data Block End Interrupt Enable Enable/Disable data block transmission end interrupt. 0: Disable 1: Enable
11	CMDACT	R/W	Command Acting Interrupt Enable Enable/Disable transmitting command interrupt. 0: Disable 1: Enable
12	TXACT	R/W	Data Transmit Acting Interrupt Enable Enable/Disable sending data interrupt. 0: Disable 1: Enable
13	RXACT	R/W	Data receive acting interrupt enable Enable/Disable receiving data interrupt. 0: Disable 1: Enable
14	TXHFERT	R/W	Tx FIFO Half Empty Interrupt Enable Enable/Disable transmit FIFO half empty interrupt. 0: Disable 1: Enable
15	RXHFFUL	R/W	Rx FIFO Half Full Interrupt Enable Enable/Disable receive FIFO half full interrupt. 0: Disable 1: Enable
16	TXFUL	R/W	Tx FIFO Full Interrupt Enable Enable/Disable transmit FIFO full interrupt. 0: Disable 1: Enable
17	RXFUL	R/W	Rx FIFO Full Interrupt Enable Enable/Disable receive FIFO full interrupt. 0: Disable 1: Enable

Field	Name	R/W	Description
18	TXEPT	R/W	Tx FIFO Empty Interrupt Enable Enable/Disable transmit FIFO empty interrupt. 0: Disable 1: Enable
19	RXFEIE	R/W	Rx FIFO Empty Interrupt Enable Enable/Disable receive FIFO empty interrupt. 0: Disable 1: Enable
20	TXDAVB	R/W	Data Available in Tx FIFO Interrupt Enable Enable/Disable transmit FIFO data available interrupt. 0: Disable 1: Enable
21	RXDAVB	R/W	Data Available in Rx FIFO Interrupt Enable Enable/Disable receive FIFO data available interrupt. 0: Disable 1: Enable
22	SDIOINTREC	R/W	SDIO Mode Interrupt Received Interrupt Enable Enable/Disable SDIO mode interrupt received. 0: Disable 1: Enable
23	ATACLPREC	R/W	CE-ATA Command Completion Signal Received Interrupt Enable Enable/Disable receiving CE-ATA command completion signal interrupt. 0: Disable 1: Enable
31:24	Reserved		

25.6.14 SDIO FIFO counter register (SDIO_FIFOCNT)

Offset address: 0x48

Reset value: 0x0000 0000

Field	Name	R/W	Description
23:0	FIFOCNT	R	Receive and Transmit FIFO Number The number of data words to be written into FIFO or read out from FIFO. Note: If the data length is not a multiple of 4, the last remaining bytes can be treated as one word.
31:24	Reserved		

25.6.15 SDIO data FIFO register (SDIO_FIFODATA)

Offset address: 0x80

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	DATA	R/W	Receive and Transmit FIFO Data Data to be written into FIFO or read out from FIFO.

26 Full-speed USBD Interface Device (USBD)

26.1 Introduction

The peripheral meets the protocol standards and technical specifications of full-speed USB2.0 interface device, and realizes the interface docking between USBD bus and APB1 bus. At the same time, the peripheral also has the functions of configurable up to 8 endpoints, synchronous transmission and endpoint double buffering mechanism, CRC check, USBD suspension/restoration operation and frame lock clock pulse generation.

26.2 Functional Description

26.2.1 Main Function

USBD module realizes the communication interface function between the MCU and the PC which conform to USB2.0 interface protocol. In the process of realizing this function, a shared private data buffer directly accessible to USBD peripheral is used. The size of this data buffer is determined by the number of configured endpoints and the maximum data packet size of different endpoints. Up to 8 bidirectional or 16 one-way endpoints can be used. Each endpoint can use 512 bytes at most. The format of data frame for USBD and PC communication is completed by hardware.

Different endpoints need to have an independent buffer description block, which is used to describe the buffer address, size and byte length of the transmitted data of the endpoint. By responding to the USBD interrupt, and judging the setting of the endpoint state register in the interrupt, the chip core enters different interrupt processing sub-functions, to process the services that the endpoint needs, and transmit corresponding types of data.

26.2.2 Explain for USBD and CAN using at the same time

Although there are actually two identical USBDs (pin was the same), but they cannot be used at the same time, so the equivalent of only one. Users could through remapping (pins' alternate functions) to implement "use at the same time", Specific instructions are as follows:

- USBD1 and CAN2 could use at the same time; USBD2 and CAN1 could use at the same time
- USBD1 and USBD2 could not use at the same time
- CAN1 and CAN2 could use at the same time

26.3 Register Address Mapping

Table 130 USBD Register Address Mapping

Register name	Description	Offset address
USBD_CTRL	USBD control register	0x40
USBD_INTSTS	USBD interrupt state register	0x44
USBD_FRANUM	USBD frame state register	0x48
USBD_ADDR	USBD device address register	0x4C
USBD_BUFFTB	USBD packet buffer description table address register	0x50
USBD_SWITCH	USBD switch register	0x100
USBD_EPn(n=[0…7])	USBD endpoint n register	From 0x00 to 0x1C
USBD_TXADDRn	Transmit buffer address register n	[USBD_ BUFFTB]+n×16
USBD_TXCNTn	Transmit data byte number register n	[USBD_ BUFFTB]+n×16+4
USBD_RXADDRn	Receive buffer address register n	[USBD_ BUFFTB]+n×16+8
USBD_RXCNTn	Receive data byte number register n	[USBD_ BUFFTB]+n×16+12

26.4 Register Functional Description

The registers of USBD module have the following three categories:

（1） General-purpose register: Interrupt register and control register;

（2） Endpoint register: Endpoint configuration register and state register;

（3） Buffer description table register: Register used to determine the packet storage address.

The base address of the buffer description table registers is specified by USBD_ BUFFTB register, and the base address of all other registers is the base address 0x4000 5C00 of USBD module. Since the APB1 bus is 32-bit addressable, the addresses of all 16-bit registers are aligned according to 32-bit word. The same address alignment method is also used for the packet buffer memory areas starting from 0x4000 6000.

26.4.1 General-purpose register

It is used to define the working mode of USBD module, interrupt processing, device address and reading of current frame number.

26.4.1.1 USBD control register (USBD_CTRL)

Offset address: 0x40

Reset value: 0x0003

Field	Name	R/W	Description
0	FORRST	R/W	Force USBD Reset Reset USBD by force, and before it is cleared by software, USBS always remains in reset state. 0: Clear 1: Reset by force
1	PWRDOWN	R/W	Power Down This mode is used to thoroughly disable the USBD module. When this bit is set to 1, the USBD module cannot be used. 0: Enable 1: Disable
2	LPWREN	R/W	Lowpower Mode Enable After this mode is turned on, when USBD is suspended, the system clock will stop or decrease to a certain frequency, and other static power consumption will be turned off except for the power supply of external pull-up resistor. Any wake-up event of USBD can clear this bit. 0: Normal mode 1: Low-power mode
3	FORSUS	R/W	Force Suspend When there is no data communication on USBD bus for 3ms, SUSP interrupt will be triggered, then the software needs to set this bit to 1. 0: Invalid 1: Pend USBD by force
4	WUPREQ	R/W	Wakeup Request Transmit the wake-up request to PC master. If this bit remains valid within 1ms to 15ms, the master will wake up the USBD. 0: Invalid 1: Valid
7:5	Reserved		
8	ESOFIEN	R/W	Expected Start of Frame Interrupt Enable 0: Disable 1: Enable
9	SOFIEN	R/W	Start of Frame Interrupt Enable 0: Disable 1: Enable
10	RSTIEN	R/W	USBD Reset Interrupt Enable 0: Disable 1: Enable
11	SUSIEN	R/W	Suspend Mode Interrupt Enable 0: Disable 1: Enable
12	WUPIEN	R/W	Wakeup interrupt Enable 0: Disable 1: Enable

Field	Name	R/W	Description
13	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable
14	PMAOUIEN	R/W	Packet Memory Area Over / Underrun Interrupt Enable 0: Disable 1: Enable
15	CTRIEN	R/W	Correct Transfer Interrupt (CTR) Enable 0: Disable 1: Enable

26.4.1.2 USBD interrupt state register (USBD_INTSTS)

Offset address: 0x44

Reset value: 0x0000

The application program confirms the interrupt request event by reading the interrupt source state information of this register.

When an interrupt event occurs, the corresponding bit will be set by the hardware. If the corresponding bit in USBD_CTRL is also set, an interrupt will be generated. After the interrupt is executed, the corresponding state bit will be cleared. It should be noted that when multiple interrupt flags are set, only one interrupt is generated.

USBD module has two-way interrupt request sources: high priority and low priority. The former interrupt request cannot be masked, while the latter can be masked. After identifying which endpoint the interrupt request is from, call the corresponding interrupt service program.

Each bit is described in detail below:

Field	Name	R/W	Description
3:0	EPID[3:0]	R	Endpoint Identifier The application program reads the processing sequence of interrupt request through software. The processing sequence is determined by the endpoint number (the smaller the endpoint number is, the higher the priority is). The priority is written by hardware.
4	DOT	R	Direction of Transaction The application program reads this bit through the software to learn the transmission direction of the pending interrupt, so as to access the corresponding operation of CTFT or CTFR bit in USBD_EPn, and this bit is written by hardware according to the transmission direction. The IN packet means that the data are transmitted from the USBD module to the PC master, while the OUT packet is the opposite. When the packet is pending, it means the data transmission is completed. 0: The CTFT bit of the corresponding endpoint is set, which means that an IN packet is pending. 1: The CTFR bit of the corresponding endpoint is set, which means that an OUT packet is pending. If the CTFT bit is set at the same time, it means that both OUT packet and IN packet are pending at the same time.
7:5	Reserved		

Field	Name	R/W	Description
8	ESOFFLG	RC_W0	Expected Start of Frame Flag When the USBD module does not receive the expected SOF packet, it is set by hardware. 0: Clear 1: Failed to receive the expected SOF packet
9	SOFFLG	RC_W0	Start of Frame Flag It indicates the start of a new USBD frame, and when the USBD module detects the SOF packet on the bus, it will be set by hardware. 0: Clear 1: SOF packet is detected
10	RSTREQ	RC_W0	USBD Reset Request The application program is readable and writable, only valid when writing 0, and set by hardware. 0: Clear 1: Reset signal is detected
11	SUSREQ	RC_W0	Suspend Mode Request It indicates a pending request from the USBD bus, which is set by hardware when there is no signal transmission on the USBD line for more than 3ms. 0: Clear 1: Pending interrupt occurred
12	WUPREQ	RC_W0	Wakeup Request If the USBD module is pending and detects a wake-up signal, it will be set by hardware. 0: Clear 1: Wake-up signal is detected
13	ERRFLG	RC_W0	Failure of Transfer Flag When NANS (master answer timeout)/CRC/BST (bit stuffing error)/FVIO (frame format error) occurs, it is set by hardware. 0: Clear 1: Transmission error occurred
14	PMOFLG	RC_W0	Packet Memory Overflow Flag When the request to access the USBD packet buffer is transmitted but the MCU does not respond for a long time, it will be set by hardware. 0: Clear 1: Packet buffer overrun occurred
15	CTFLG	R	Correct Transfer Flag This bit is set by hardware after the endpoint completes data transmission once correctly. 0: Meaningless 1: Data transmission is completed correctly

26.4.1.3 USBD frame state register (USBD_FRANUM)

Offset address: 0x48

Reset value: 0x0XXX; X means undefined value

Field	Name	R/W	Description
10:0	FRANUM[10:0]	R	Frame Sequence Number

Field	Name	R/W	Description
			Record the 11-bit frame code of the latest SOF packet, and the serial number will automatically increase as the quantity of transmit frames of the master increases.
12:11	LSOFNUM[1:0]	R	Continuous Lost SOF Number Record the number of SOF packets lost in the start of consecutive frame, and this bit is cleared when the SOF packet is received again.
13	LOCK	R	Lock If USBD continuously detects two or more SOP packets, it will be set by hardware, then the frame counter will stop counting until the USBD is reset or the bus is pending.
14	RXDMSTS	R	Receive Data- line Status It indicates the state of USBD D- (Data minus) data line, and can detect whether the wake-up condition appears in pending state.
15	RXDPSTS	R	Receive Data+ Line Status It indicates the state of USBD D+ (Data plus) data line, and can detect whether the wake-up condition appears in pending state.

26.4.1.4 USBD device address register (USBD_ADDR)

Offset address: 0x4C

Reset value: 0x0000

Field	Name	R/W	Description
6:0	ADDR[6:0]	R/W	Device Address Record the address value assigned by USBD master for the device in the process of enumeration.
7	USBDEN	R/W	USBD Enable Enable USBD. 0: Disable 1: Enable
15:8	Reserved		

26.4.1.5 USBD packet buffer description table address register (USBD_BUFFTB)

Offset address: 0x50

Reset value: 0x0000

Field	Name	R/W	Description
2:0	Reserved		
15:3	BUFFTB	R/W	Buffer Table Record the start address of the packet buffer description table.
31:16	Reserved		

26.4.1.6 USBD switch register (USBD_SWITCH)

Offset address: 0x100

Reset value: 0x0000_0000

Field	Name	R/W	Description
0	SWITCH[0]	R/W	USBD Switch The default is 0. 0: USBD1 is used 1: USBD2 is used; then it can be used with CAN at the same time (Pins PA11 and PA12 are used by USBD).
15:1	Reserved		

26.4.2 USBD endpoint register

Each endpoint has corresponding USBD_EPn (n=0…7) registers to store various state information of the endpoint.

26.4.2.1 USBD endpoint n register (USBD_EPn) (n=0 to 7)

Offset address: 0x00 to 0x1C

Reset value: 0x0000

When the endpoint register receives USBD bus reset or the FORRST of USBD_CTRL is set, except that CTFT and CTFR bits remain unchanged for data transmission, other bits will be reset.

Each bit is described in detail below:

Field	Name	R/W	Description
3:0	ADDR[3:0]	R/W	Endpoint Address Before an endpoint is enabled, the application program must set these four bits as an address.
5:4	TXSTS	T	Status Bits for Transmission Transfers This bit is used to identify the current state of the endpoint, and the transmitting state code table indicates all states. The application program is readable and writable; write 0 and this bit is invalid; write 1 to flip this bit.
6	TXDTOG	T	Data Toggle for Transmission Transfers (1) For asynchronous endpoints, it is used to indicate the Toggle bit of the next data packet to be transmitted: 0：DATA0 1：DATA1 (2) For double-buffer endpoints, it can be used for packet buffer exchange. (3) For synchronous endpoints, as only DATA0 is transmitted, this bit is only used to support packet buffer exchange. The hardware sets the bit after receiving the data packet. The application program is readable and writable; write 0 and this bit is invalid; write 1 to flip this bit.
7	CTFT	RC_W0	Correct Transfer Flag for Transmission This bit is set by hardware after a correct IN packet is transmitted. 0: Clear this bit and it is meaningless to read out 1: IN packet is transmitted correctly. Writing is invalid The application program is readable and writable; write 0 and this bit is valid; write 1 and this bit is invalid.
8	KIND	R/W	Endpoint Kind

Field	Name	R/W	Description
			Please refer to the definition table of special types of endpoints for specific definitions.
10:9	TYPE	R/W	Endpoint Type Please refer to the definition table of endpoint types for specific definitions.
11	SETUP	R	Setup Transaction Completed (read-only) It indicates whether the packet that currently completes transmission is SETUP packet. After the USBD module receives a correct SETUP packet, it is set by hardware. 0: No 1: Yes Note: This bit can be modified only when CTFR is 0, and cannot be modified when CTFR is 1.
13:12	RXSTS	T	Status Bits for Reception Transfers This bit is used to identify the current state of the endpoint, and the receiving state code table indicates all states. The application program is readable and writable; write 0 and this bit is invalid; write 1 to flip this bit.
14	RXDTOG	T	Data Toggle for Reception Transfers (1) For asynchronous endpoints, flag the Toggle bit of the next data packet that is expected to be received, and it is set by hardware: 0：DATA0 1：DATA1 (2) For control endpoint, after receiving the SETUP packet, it is cleared by hardware. (3) For double-buffer endpoints, it is used for the exchange of double buffers. (4) For synchronous endpoints, it is only used for the exchange of double buffers, without flipping. The application program is readable and writable; write 0 and this bit is invalid; write 1 to flip this bit.
15	CTFR	RC_W0	Correct Transfer Flag for Reception This bit is set by hardware after OUT or SETUP packet is received correctly. 0: Clear 1: Packet is received correctly The application program is readable and writable; write 0 and this bit is valid; write 1 and this bit is invalid.

Table 131 Receive State Codes

RXSTS	Description
00	DISABLED: All receiving requests are ignored by the endpoint.
01	STALL: The endpoint responds to all receive requests with STALL packet.
10	NAK: The endpoint responds to all receive requests with NAK packet.
11	VALID: The endpoint can be used for receiving.

Table 132 Endpoint Type Codes

TYPE[1:0]	Description
00	BULK: Bulk endpoint
01	CONTROL: Control endpoint
10	ISO: Synchronous endpoint
11	INTERRUPT: Interrupt endpoint

Table 133 Definition of Special Types of Endpoints

TYPE	Meaning
00（BULK）	DBL_BUF: Double-buffer endpoint
01（CONTROL）	STATUS_OUT
Other	Unused

Table 134 Transmit State Codes

TXSTS	Description
00	DISABLED: All transmit requests are ignored by the endpoint.
01	STALL: The endpoint responds to all transmit requests with STALL packet.
10	NAK: The endpoint responds to all transmit requests with NAK packet.
11	VALID: The endpoint can be used for transmitting.

26.4.3 Buffer register

The buffer description table is located in the packet buffer to configure the address and size of the packet buffer shared by the USBD module and the MCU core.

26.4.3.1 Transmit buffer address register n (USBD_TXADDRn) (n=0 to 7)

Offset address: [USBD_BUFFTB]+n×16

Local address of USBD: [USBD_BUFFTB]+n×8

Field	Name	R/W	Description
0	Reserved, must be 0.		
15:1	ADDR	R/W	Transmission Buffer Address Record the starting address of the buffer where the data to be transmitted is located when the next IN packet is received.

26.4.3.2 Transmit data byte number register n (USBD_TXCNTn) (n=0 to 7)

Offset address: [USBD_BUFFTB]+n×16+4

Local address of USBD: [USBD_BUFFTB]+n×8+2

Field	Name	R/W	Description
9:0	CNT	R/W	Transmission Byte Count This bit records the data byte count to be transmitted when receiving next IN group.

Field	Name	R/W	Description
15:10	Reserved		

Note: The double buffer and synchronous IN endpoint have two USBD_TXCNT registers: USBD_TXCNT_1 and USBD_TXCNT_0, and the content is as follows:

Field	Name	R/W	Description
9:0	CNT_0	R/W	Transmission Byte Count0 Record the data byte count to be transmitted when receiving next IN group.
15:10	Reserved		
25:16	CNT_1	R/W	Transmission Byte Count1 Record the data byte count to be transmitted when receiving next IN group.
31:26	Reserved		

26.4.3.3 Receive buffer address register n (USBD_RXADDRn) (n=0 to 7)

Offset address: [USBD_BUFFTB]+n×16+8

Local address of USBD: [USBD_BUFFTB]+n×8+4

Field	Name	R/W	Description
0	Reserved, must be 0.		
15:1	ADDR	R/W	Reception Buffer Address Record the starting address of the buffer used to save data when receiving the next OUT or SETUP packet.

26.4.3.4 Receive data byte register n (USBD_RXCNTn) (n=0 to 7)

Offset address: [USBD_BUFFTB]+n×16+12

Local address of USBD: [USBD_BUFFTB]+n×8+6

It is used to store two parameters that need to be used when receiving packets: the buffer size of receiving packets and the number of bytes actually received by USBD.

Field	Name	R/W	Description
9:0	CNT	R	Reception Byte Count It is written by the USBD module to record the actual number of bytes of the latest OUT or SETUP packet received by the endpoint.
14:10	BLKNUM	R/W	Number of Blocks Record the number of memory blocks allocated, which determines the size of the packet buffer.
15	BLKSIZE	R/W	Block Size Determine the size of memory block, which determines the buffer size. 0: 2 bytes 1: 32 bytes

Note: The double buffer and synchronous IN endpoint have two USBD_RXCNT registers: USBD_RXCNT_1 and USBD_RXCNT_0, and the content is as follows:

Field	Name	R/W	Description
9:0	CNT_0[9:0]	R	Reception Byte Count Record the actual number of bytes of the latest OUT or SETUP group received by the endpoint.
14:10	BLKNUM_0	R/W	Number of Blocks Record the number of memory blocks allocated, which determines the size of the packet buffer.
15	BLKSIZE_0	R/W	Block Size Determine the size of memory block. 0: 2 bytes 1: 32 bytes
25:16	CNT_1	R	Reception Byte Count Record the actual number of bytes of the latest OUT or SETUP group received by the endpoint.
30:26	BLKNUM_1	R/W	Number of Blocks Record the number of memory blocks allocated.
31	BLKSIZE_1	R/W	Block Size Determine the size of memory block. 0: 2 bytes 1: 32 bytes

Table 135 Definition of Packet Buffer Sizes

Value of BLKNUM[4:0]	When BLKSIZE=0 Size of packet buffer	Size of packet buffer when BLKSIZE=1
00000	Not allowed to use	32 byte
00001	2 bytes	64 bytes
00010	4 bytes	96 bytes
00011	6 bytes	128 bytes
…	…	…
01111	30 bytes	512 bytes
10000	32 bytes	Reserved
10001	34 bytes	Reserved
10010	36 bytes	Reserved
…	…	…
11110	60 bytes	Reserved
11111	62 bytes	Reserved

27 Analog-to-digital Converter (ADC)

27.1 Full Name and Abbreviation Description of Terms

Table 136 Full Name and Abbreviation Description of ADC Terms

Full name in English	English abbreviation
Analog watchdog	AWD
Conversion	C
Injected	INJ
Regular	REG
Start	S
Scan	SCAN
Single	SINGLE
Automatic	A
Group	G
Discontinuous	DISC
Count	CNT
Dual	DUAL
Continuous	C
Calibration	CAL
Reset	RST
Alignment	ALIGN
External	EXT
Event	E
Trigger	TRG
Temperature	T
Sensor	S
Time	TIM
Sample	SMP
Offset	OF
High	H
Low	L
Threshold	T
Sequence	SEQ

Full name in English	English abbreviation
Length	LEN
Regular Channels	REG
Injected Channel	INJ
Injected Group	INJG
Automatic	A
Conversion	C
Analog Watchdog	AWD
Discontinuous Mode	DISC
Scan Mode	SCAN
Continuous Conversion	CONTC
Single Conversion	SINGLEC
External	EXT
External Trigger	EXTTRG
Sample Time	SMPTIM
Sequence	SEQ
Number	NUM

27.2 Introduction

Series products have three ADCs with 12-bit accuracy, and each ADC has up to 16 external channels and 2 internal channels. ADC1 and ADC2 have 16 external channels respectively, ADC3 generally has 8 external channels, the A/D conversion mode of each channel has single, continuous, scan or discontinuous modes, and the ADC conversion results can be stored in 16-bit data register by left alignment or right alignment.

27.3 Main Characteristics

（1） ADC power supply requirements: From 2.4V to 3.6V; the general power supply voltage is 3.3V.

（2） ADC input range: $V_{REF-} \leq V_{IN} \leq V_{REF+}$.

（3） 12-bit resolution

（4） ADC conversion time

- Formula: TCONV=samplting time+12.5 cycles

- The sampling time is controlled by SMPCYCCFGx[2:0] bit, and the minimum sampling cycle is 1.5; when ADCCLK=14MHz, the sampling time is 1.5 cycles: TCONV=1.5 cycles +12.5 cycles=14 cycles=1 μs.

（5） Mode input channel category

- External GPIO input channel
- One internal temperature sensor (V_{SENSE}) input channel
- One internal reference voltage (V_{REFINT}) input channel

（6） Channel conversion mode

- Single channel conversion mode: single conversion mode, continuous conversion mode
- Input channel classification: regular channel, injected channel
- One-group channel conversion mode: scan mode, discontinuous mode and injected channel management
- ADC mode: Independent ADC mode, and dual ADC mode

（7） Trigger mode

- On-chip timer signal trigger
- External pin signal trigger
- Software trigger

（8） Data register

- Regular data register
- Injected data register

（9） Interrupt

- End of conversion interrupt
- Analog watchdog interrupt

（10） DMA request supporting regular data conversion

（11） Data alignment

- Configurable data alignment of DALIGNCFG bit of data register ADC_CTRL2 is left or right alignment.

（12） Self-calibration

- Enable calibration by setting CAL bit of ADC_CTRL2 register. CAL bit is set to 1 during calibration and is cleared by hardware after calibration; calibration shall be performed every time before power-on.

27.4 Functional Description

27.4.1 ADC Pins

Table 137 ADC Pins

Name	Instruction	Signal type
V_{REF+}	High-end/Positive electrode reference voltage used by ADC, 2.4V≤V_{REF+}≤V_{DDA}	Input, analog reference positive electrode
V_{DDA}(1)	Equivalent to analog power supply of V_{DD} and: 2.4V≤V_{DDA}≤V_{DD}(3.6V)	Input, analog power supply
V_{REF-}	Low-end/Negative electrode reference voltage used by ADC, V_{REF-} =V_{SSA}	Input, analog reference negative electrode
V_{SSA}(1)	Equivalent to analog power supply of V_{ss}	Input, analog power ground
ADCx_IN[15:0]	16 analog input channels	Analog input signal

Note: 1. V_{DDA} and V_{SSA} should be connected to V_{DD} and V_{SS} respectively.

27.4.2 ADC Conversion Mode

The product has multiple built-in ADCs and channels (refer to the *Data Manual* for the specific number), which can be combined into a variety of conversion modes.

Multiple built-in ADCs; according to the number of ADCs, the conversion mode can be classified into independent ADC mode and dual ADC mode; multiple built-in channels, which can be classified into two groups, namely regular channel and injected channel. The internal conversion mode of each group can be divided into scan mode and discontinuous mode; for the internal channels of each group, the conversion mode is divided into single conversion mode and continuous conversion mode.

In the application, according to the actual application requirements, the number of ADC, the number of conversion channels and the conversion mode of each channel, the ADC conversion mode meeting the requirements can be designed.

27.4.2.1 Conversion mode of single ADC and single channel

Single ADC channel

Single ADC and single channel are not enabled by external trigger software. The conversion mode is single and continuous concurrent disabling of scan. The result of data conversion is right alignment. After the single ADC conversion is completed, the interrupt is triggered, and data is read in the interrupt service function, not using DMA transmission.

Single conversion mode

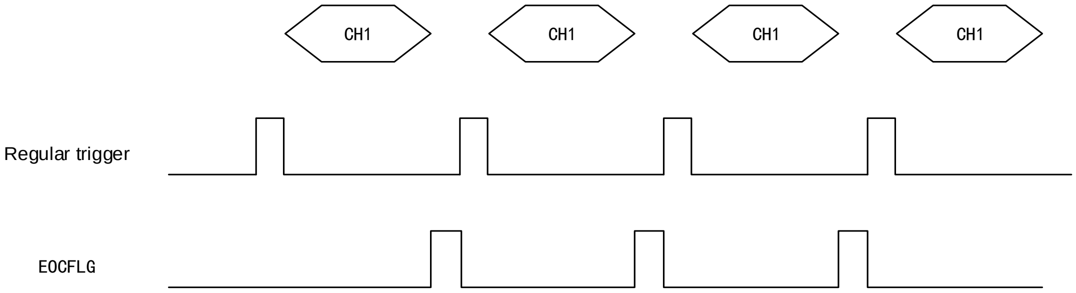
In this mode, for single channel, only one conversion is performed for this channel, and for multiple channels, only one conversion is performed for this group of channels .

This mode is started by the ADCEN bit of configuration register ADC_CTRL2 or is started by external trigger.

After one conversion of regular channel is over, the converted data will be stored in 16-bit ADC_REGDATA register, and EOCFLG bit will be set to 1. If configuration EOCIEN bit is set to 1, an interrupt will be generated.

After one conversion of injected channel is over, the converted data will be stored in 16-bit ADC_INJDATA1 register, and INJEOCFLG bit will be set to 1. If configuration INJEOCIEN bit is set to 1, an interrupt will be generated.

Figure 127 Single Conversion Mode Timing Diagram



Continuous conversion mode

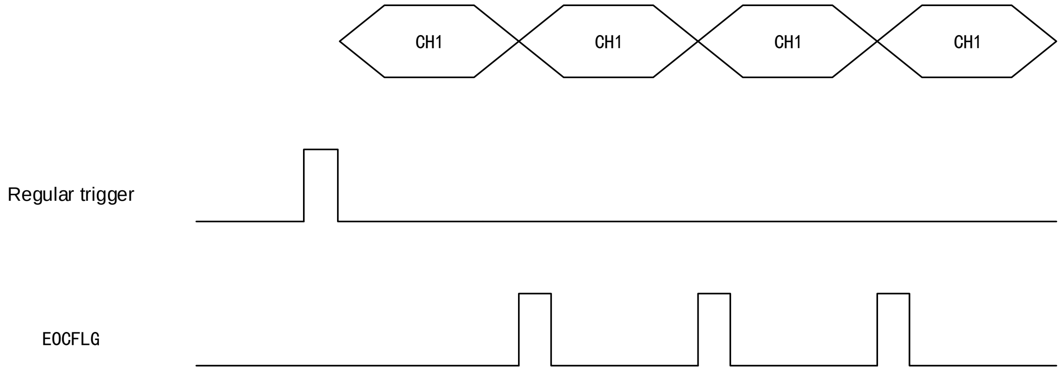
In this mode, for single channel, continuous conversion is conducted for this channel.

This mode is started by the ADCEN bit of configuration register ADC_CTRL2 or is started by external trigger.

After the conversion of one regular channel is over, the converted data will be stored in 16-bit ADC_REGDATA register, and EOCFLG bit will be set to 1. If configuration EOCIEN bit is set to 1, an interrupt will be generated.

After the conversion of one injected channel is over, the converted data will be stored in 16-bit ADC_INJDATA1 register, and INJEOCFLG bit will be set to 1. If configuration INJEOCIEN bit is set to 1, an interrupt will be generated.

Figure 128 Continuous Conversion Mode Timing Diagram



27.4.2.2 Conversion mode of single ADC and one group of channels

Single ADC and multiple channels

Enable the scan mode under single-ADC multi-channel condition, the conversion is triggered by software rather than externally, the result of data conversion is right aligned, and the data of ADC conversion results are transmitted to the memory by DMA.

Classification of analog input channels

Regular channel group

- The regular group consists of 16 channels
- Regular channel conversion sequence is determined by the register ADC_REGSEQx
- The total number of conversion channels of regular group is determined by REGSEQLEN[3:0] bit of configuration register ADC_REGSEQ1

Injected channel group

- The injected group consists of 4 channels
- Injected channel conversion sequence is determined by the register ADC_INJSEQ
- The total number of conversion channels of injected group is determined by INJSEQLEN[1:0] bit of configuration register ADC_INJSEQ

Internal input channel

Temperature sensor:

- The temperature sensor is used to measure the internal temperature of the chip
- The temperature sensor selects ADC1_IN16 input channel

- Start through TSVREFEN bit of the configuration register ADC_CTRL2

Internal reference voltage V_{REFINT}:

- The internal reference voltage is used to provide a stable voltage output for ADC
- Internal reference voltage V_{REFINT} is used to select ADC1_IN17 input channel

Channel conversion sequence

Configuration of regular sequence registers:

- Configure REGSEQC1[4:0]~REGSEQC6[4:0] bits of the register ADC_REGSEQ3 to set No. 1~6 conversion channels
- Configure REGSEQC7[4:0]~REGSEQC12[4:0] bits of the register ADC_REGSEQ3 to set No. 7~12 conversion channels
- Configure REGSEQC13[4:0]~REGSEQC16[4:0] bits of the register ADC_REGSEQ1 to set No. 13~16 conversion channels
- Configure REGSEQLEN[3:0] of the register ADC_REGSEQ1 to set the number of channels for conversion

Configuration of injected sequence register:

- Configure NJSEQC1[4:0]~INJSEQC4[4:0] bit of the register ADC_INJSEQ to set No. 1~4 conversion channels
- Configure INJSEQLEN[1:0] of the register ADC_INJSEQ to set the number of channels for conversion
- If the value of INJSEQLEN is less than 4, the conversion sequence will be different and start from (4-INJSEQLEN).

Channel conversion mode

Scan Mode

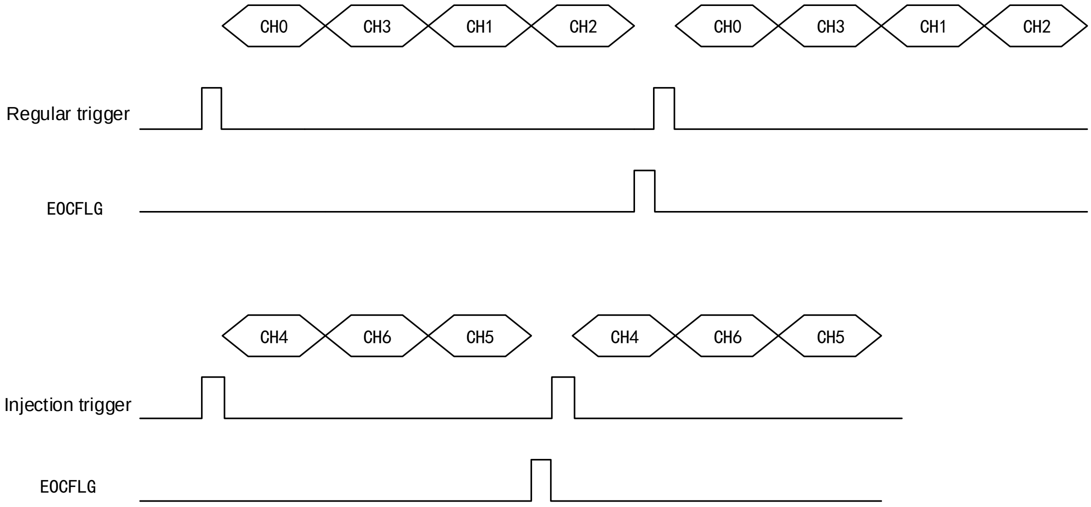
This mode is applicable to one group of channels, which is equivalent to a single conversion on each channel of one group of channels.

This mode is started by SCANEN bit of configuration register ADC_CTRL1, and after startup, ADC scans all channels which are arranged according to the sequence register ADC_REGSEQ or the ADC_INJSEQ, and after each channel conversion is completed, it will be automatically converted to the next channel of the group.

If the configuration CONTCEN bit is set to 1, the conversion will continue from the first channel of the group when the last channel of the group completes conversion.

If the configuration DMAEN bit is set to 1, the DMA controller will transmit the converted data of regular channel to SRAM every time the channel conversion is completed.

Figure 129 Scan Mode Timing Diagram



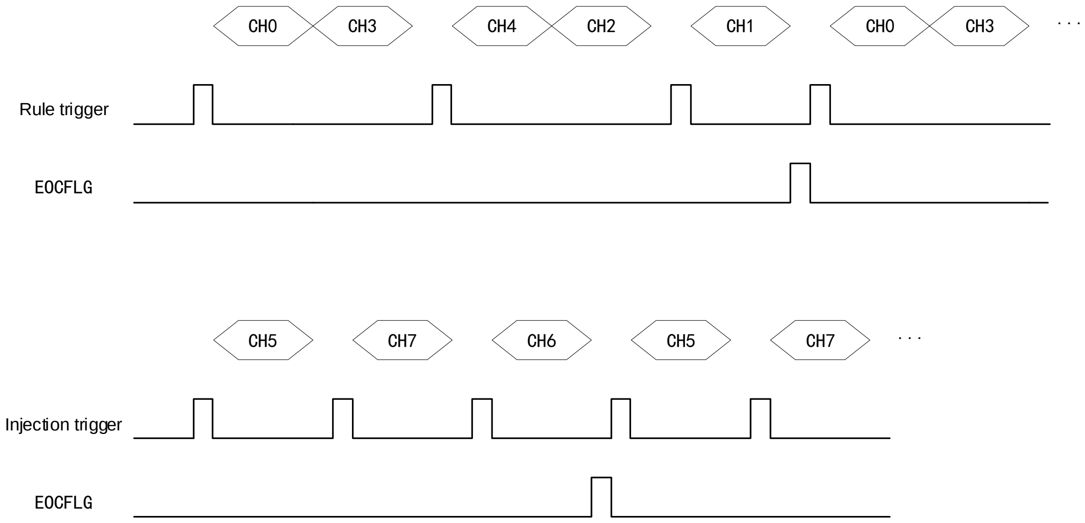
Discontinuous mode

This mode is suitable for a group of channels, which is equivalent to continuous conversion of multiple channels in a group of channels.

For regular groups, this mode is started by REGDISCEN bit of configuration register ADC_CTRL1; after startup, conduct short sequence conversion of n channels (n<=8), and n is determined by DISCNUMCFG[2:0] of configuration register ADC_CTRL1; next round of conversion of n channels can be started through software control or external trigger source and when the conversion of all channels of this group is completed, EOCFLG bit will be set to 1.

For injected groups, this mode is enabled by INJDISCEN bit of configuration register ADC_CTRL1; after startup, one channel will be converted according to the configuration sequence of the sequence register; conversion of next channels can be started by software control or external trigger source and when the conversion of all channels of this group is completed, EOCFLG bit and INJEOCFLG bit will be set to 1.

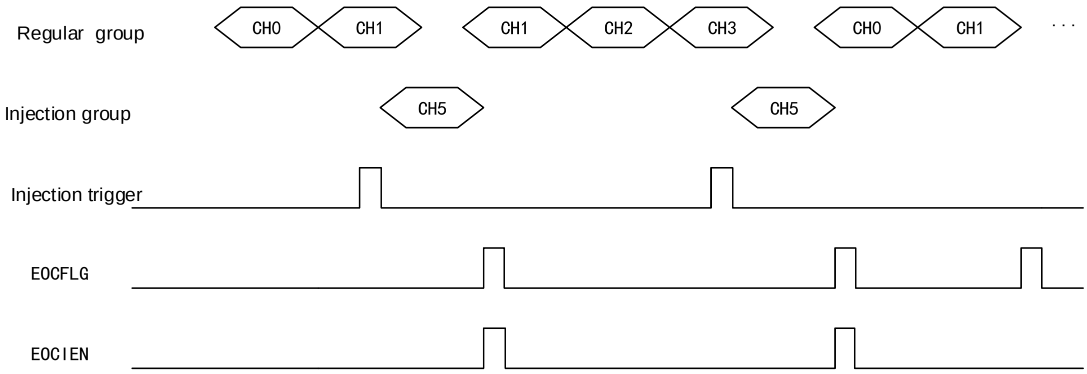
Figure 130 Discontinuous Mode Timing Diagram



Injected channel management

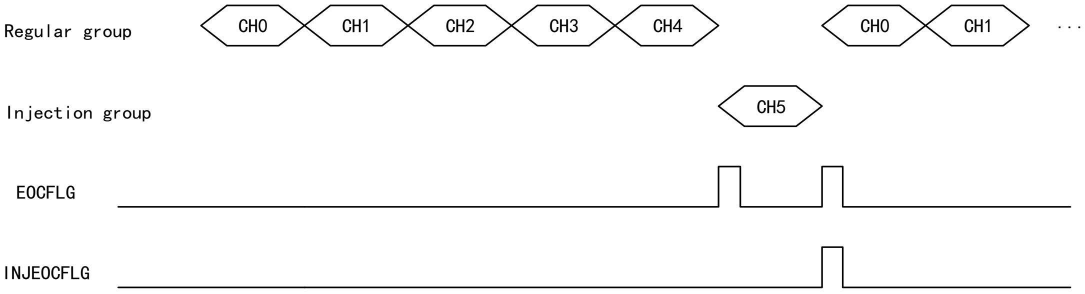
Trigger injection: Start by clearing INJGACEN bit of the register ADC_CTRL1 and configuring the SCANEN bit. If a software trigger or external trigger is generated during the conversion of regular group channels, the injected conversion will be triggered. At this time, the regular channel conversion will stop, the injected channel sequence will start conversion, and after the injected group channel conversion is completed, the regular group channel conversion will be recovered.

Figure 131Trigger Injection Timing Diagram



Automatic injection: Start by INJGACEN bit of configuration register ADC_CTRL1; after conversion of the regular group channels is completed, the injected group channels will start conversion automatically; in the automatic injection mode, external trigger of the injected group channels must be disabled; if the CONTCEN bit of the register ADC_CTRL2 is also configured, all channels of regular group and injection group will convert continuously.

Figure 132 Automatic Injection Timing Diagram



27.4.2.3 Conversion mode of dual-ADC and one group of channels

For products with two or more ADC modules, dual ADC mode is used. ADC1 is the master ADC by default, while others are the slave ADC by default, and dual ADC mode is set by configuring DUALMCFG[2:0] bit in ADC1_CTRL1 register.

When the configuration is triggered by external event, it is required to set it to trigger only the master ADC, and then configure the slave ADC to be triggered by software. (External trigger of master and slave must occur at the same time)

There are eight possible dual-ADC modes:

（1） **Simultaneous regular mode**

The simultaneous regular mode means ADC1 and ADC2 convert a regular channel group at the same time. Two ADCs cannot convert one channel at the same time.

The external trigger event is determined by REGEXTTRGSEL[2:0] of the register ADC1_CTRL2.

After ADC conversion, DMA transmission request will be generated, and converted data of ADC1 are stored in low 16 bits of the register ADC1_REGDATA, while ADC2 converted data are stored in high 16 bits of the register ADC1_REGDATA.

EOCFLG interrupt will be generated after all ADC regular channels are converted.

（2） **Simultaneous injection mode**

The simultaneous injection mode means ADC1 and ADC2 convert one injected channel group at the same time. Two ADCs cannot convert one channel at the same time.

The external trigger event is determined by INJGEXTTRGSEL[2:0] of the register ADC1_CTRL2.

After ADC conversion is over, the converted data will be stroed in the register ADC_INJDATAx.

INJEOCFLG interrupt will be generated after all ADC injected channels are converted.

（3） **Fast cross mode**

The fast cross mode means ADC1 and ADC2 collect a regular channel group alternately, with a short interval time.

The external trigger event is determined by REGEXTTRGSEL[2:0] of the register ADC_CTRL2; after the trigger is generated, ADC2 will be started, and ADC1 will be started after delay of seven ADC clock cycles.

The sampling time shall be less than seven ADC clock cycles.

（4） **Slow cross mode**

The slow cross mode means ADC1 and ADC2 collect a regular channel group alternately, with a long interval time.

The external trigger event is determined by REGEXTTRGSEL[2:0] of the register ADC_CTRL2; after the trigger is generated, ADC2 will be started, and ADC1 will be started after delay of 14 ADC clock cycles.

The sampling time shall be less than 14 ADC clock cycles.

（5） **Alternate trigger mode**

The alternate trigger mode means ADC1 and ADC2 collect the injected channel group by turns.

The external trigger event is determined by INJGEXTTRGSEL[2:0] of the register ADC1_CTRL2; after the trigger is generated, ADC1 will start conversion, and after all channels are converted, ADC2 will start conversion.

If discontinuous mode is enabled for ADC1 and ADC2, after triggered, ADC1 will start converting the first injected channel; after the second trigger is generated, ADC2 will start to convert the first injected channel and so on.

（6） **Mixed simultaneous regular/injection mode**

The mixed simultaneous regular/injection mode means after the simultaneous regular mode is interrupted, the simultaneous injection mode will be enabled.

In this mode, a sequence of the same length must be converted or a trigger interval time must be set to complete the conversion of a longer sequence.

（7） **Mixed simultaneous regular + alternate trigger mode**

The mixed simultaneous regular + alternate trigger mode means after the simultaneous regular mode is interrupted, the alternate trigger mode will be enabled.

In this mode, a sequence of the same length must be converted or a trigger interval time must be set to complete the conversion of a longer sequence.

（8） **Mixed simultaneous injection + cross mode**

The mixed simultaneous injection + cross mode means after the cross regular mode is interrupted, the simultaneous injection mode will be enabled.

27.4.3 External Trigger

Register configuration of external trigger is as follows:

- The external event trigger of regular group channel is enabled by REGEXTTRGSEL[2:0] bit of configuration register ADC_CTRL2
- The external event trigger of injected group channel is started by INJGEXTTRGSEL[2:0] bit of configuration register ADC_CTRL2.

Table 138 External Trigger of Regular Channel of ADC1 and ADC2

Trigger source	REGEXTTRGSEL[2:0]	Trigger type
TMR1_CC1	000	Internal signal from on-chip timer
TMR1_CC2	001	
TMR1_CC3	010	
TMR2_CC2	011	
TMR3_TRGO	100	
TMR4_CC4	101	
EINT Line 11/TMR8_TRGO	110	External pin/internal signal from on-chip timer
REGSWSC	111	Software control bit

Table 139 External Trigger of Injected Channel of ADC1 and ADC2

Trigger source	INJGEXTTRGSEL[2:0]	Trigger type
TMR1_TRGO	000	Internal signal from on-chip timer
TMR1_CC4	001	
TMR2_TRGO	010	
TMR2_CC1	011	
TMR3_CC4	100	
TMR4_TRGO	101	
EINT Line 15/TMR8_CC4	110	External pin/internal signal from on-chip timer
INJSWSC	111	Software control bit

Table 140 External Trigger of Regular Channel of ADC3

Trigger source	REGEXTTRGSEL[2:0]	Trigger type
TMR3_CC1	000	Internal signal from on-chip timer

Trigger source	REGEXTTRGSEL[2:0]	Trigger type
TMR2_CC3	001	
TMR1_CC3	010	
TMR8_CC1	011	
TMR8_TRGO	100	
TMR5_CC1	101	
TMR5_CC3	110	
REGSWSC	111	Software control bit

Table 141 External Trigger of Injected Channel of ADC3

Trigger source	REGEXTTRGSEL[2:0]	Trigger type
TMR1_TRGO	000	Internal signal from on-chip timer
TMR1_CC4	001	
TMR4_CC3	010	
TMR8_CC2	011	
TMR8_CC4	100	
TMR5_TRGO	101	
TMR5_CC4	110	
INJSWSC	111	Software control bit

27.4.4 Data Register

27.4.4.1 Regular data register

ADC_REGDATA is a 32-bit ADC regular data register. In single-ADC mode, only the lower 16 bits are used to store the converted data. In dual-ADC mode, the lower 16 bits are used to store the converted data of ADC1 while the higher 16 bits are used to store the converted data of ADC2. The data are left aligned or right aligned.

It is determined by DALIGNCFG bit of configuration register ADC_CTRL2 whether to use DMA transmission. There are at most 16 regular channels, but only one regular data register. Therefore, data coverage will occur in multi-channel conversion, and DMA transmission is needed at this time.

27.4.4.2 Injection data memory

ADC_INJDATAx (x=1,2,3,4) is ADC injected data register, and there are four 32-bit registers, of which the low 16 bits are effective and the high 16 bits are reserved. There are at most four injected channels and four injection data registers, so data coverage will not occur in multi-channel conversion. The data are left aligned or right aligned.

27.4.5 Interrupt

27.4.5.1 End of conversion interrupt

Interrupt of end of conversion of regular group channels

An interrupt will be generated by the end of conversion of regular channels; read the value of the regular data register in the interrupt function.

Determine by EOCFLG bit of configuration register ADC_STS.

Interrupt of end of conversion of injected group channels

An interrupt will be generated after the conversion of injected channels is completed; read the value of the regular data register in the interrupt function.

Determine by INJEOCFLG bit of configuration register ADC_STS.

27.4.5.2 Analog watchdog interrupt

If the input analog voltage is not within the threshold range, an analog watchdog interrupt will be generated.

Determine by configuring AWDFLG bit of the register ADC_STS.

27.4.6 DMA

DMA request will be generated after the conversion of regular channels is completed; the converted data result can be transmitted to the memory from the ADC_REGDATA register.

Only ADC1 and ADC3 can generate DMA request, and the conversion results of ADC2 are transmitted through the DMA function of ADC1.

27.5 Register Address Mapping

Table 142 ADC Register Address Mapping

Register name	Description	Offset address
ADC_STS	ADC state register	0x00
ADC_CTRL1	ADC control register 1	0x04
ADC_CTRL2	ADC control register 2	0x08
ADC_SMPTIM1	ADC sampling time register 1	0x0C
ADC_SMPTIM2	ADC sampling time register 2	0x10
ADC_INJDOFx	ADC injected channel data offset register x	0x14-0x20
ADC_AWDHT	Analog watchdog high-threshold register	0x24
ADC_AWDLT	Analog watchdog low-threshold register	0x28

Register name	Description	Offset address
ADC_REGSEQ1	ADC regular sequence register 1	0x2C
ADC_REGSEQ2	ADC regular sequence register 2	0x30
ADC_REGSEQ3	ADC regular sequence register 3	0x34
ADC_INJSEQ	ADC injected sequence register	0x38
ADC_INJDATAx	ADC injected data register X	0x3C–0x48
ADC_REGDATA	ADC regular data register	0x4C

27.6 Register Functional Description

27.6.1 ADC state register (ADC_STS)

Offset address: 0x00

Reset value: 0x0000 000

Field	Name	R/W	Description
0	AWDFLG	RC_W0	Analog Watchdog Occur Flag This bit is set to 1 by hardware and cleared by software, indicating whether an analog watchdog event occurs. 0: No occurrence 1: Occurred
1	EOCFLG	RC_W0	End Of Conversion Flag 0: Not completed 1: Completed
2	INJEOCFLG	RC_W0	Injected Channel End Of Conversion Flag 0: Not completed 1: Completed
3	INJCSFLG	RC_W0	Injected Channel Conversion Start Flag 0: Not start 1: Start
4	REGCSFLG	RC_W0	Regular Channel Conversion Start Flag 0: Not start 1: Start
31:5	Reserved		

27.6.2 ADC control register 1 (ADC_CTRL1)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	AWDCHSEL	R/W	Analog Watchdog Channel Select 00000: ADC analog input channel 0 00001: ADC analog input channel 1 …… 01111: ADC analog input channel 15 10000: ADC analog input channel 16

Field	Name	R/W	Description
			10001: ADC analog input channel 17 Other value: Reserved For this register, pay attention to the followings: (1) The analog input channel 16 and channel 17 of ADC1 are connected to the temperature sensor and V_{REFINT} in the chip respectively (2) The analog input channel 16 and channel 17 of ADC2 are connected to V_{SS} in the chip; (3) ADC3 analog input channels 9, 14, 15, 16 and 17 are connected to V_{ss}.
5	EOCIEN	R/W	EOC Interrupt Enable Used to enable the generation of interrupt after the conversion is completed. 0: Disable 1: Enable
6	AWDIEN	R/W	Analog Watchdog Interrupt Enable If the bit is set and in scan mode, when the watchdog detects that the value exceeds the threshold, an interrupt will be generated and the scan will be aborted. 0: Disable 1: Enable
7	INJEOCIEN	R/W	Interrupt Enable For Injected Channels End Of Conversion Flag 0: Disable 1: Enable
8	SCANEN	R/W	Scan Mode Enable In the scan mode, convert the channel selected by ADC_REGSEQX or ADC_INJSEQX register. 0: Disable 1: Enable Note: If EOCINTEN or INJEOCINTEN bit is set respectively, EOC or INJEOC interrupt will be generated only after the last channel is converted.
9	AWDSGLEN	R/W	Enable The Watchdog On A Single Channel In Scan Mode This channel is specified by AWDCHSEL[4:0] bit. 0: Enable on all channels 1: Enable on a single channel;
10	INJGACEN	R/W	Automatic Injected Group Conversion Enable Used to enable automatic conversion of injected channels after the conversion of regular channel group is completed. 0: Disable 1: Enable
11	REGDISCEN	R/W	Discontinuous Mode On Regular Channels Enable 0: Disable 1: Enable
12	INJDISCEN	R/W	Discontinuous Mode On Injected Channels Enable 0: Disable 1: Enable

Field	Name	R/W	Description
15:13	DISCNUMCFG	R/W	Discontinuous Mode Channel Number Configure 000: One channel 001: Two channels …… 111: Eight channels
19:16	DUALMCFG	R/W	Dual ADC Mode Configure 0000: Independent mode 0001: Mixed synchronous regular + injected synchronous mode 0010: Mixed simultaneous regular + alternate trigger mode 0011: Mixed simultaneous injection + fast cross mode 0100: Mixed simultaneous injection + slow cross mode 0101: Injected simultaneous mode 0110: Regular simultaneous mode 0111: Fast cross mode 1000: Slow cross mode 1001: Alternate trigger mode Others: Reserved In ADC2 and ADC3, these bits are reserved bits; in dual ADC mode, changing the channel configuration will result in a restart condition, which will result in loss of synchronization. It is recommended to turn off dual ADC mode (i.e. configure as independent mode) before making any configuration changes.
21:20	Reserved		
22	INJAWDEN	R/W	Enable the Analog Watchdog Function On the Injected Channels 0: Disable 1: Enable
23	REGAWDEN	R/W	Enable the Analog Watchdog Function On the Regular Channels 0: Disable 1: Enable
31:24	Reserved		

27.6.3 ADC control register 2 (ADC_CTRL2)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ADCEN	R/W	ADC Enable (1) If this bit is set to 0, write 1 to power on ADC and start ADC conversion (2) If this bit is set to 1, write 1 to start conversion 0: Disable ADC conversion and calibration and enter the power-down mode 1: Enable ADC and start conversion Note: To prevent triggering wrong conversion, if another bit and this bit in this register are changed, conversion will not be triggered.
1	CONTCEN	R/W	Continuous Conversion Mode Enable

Field	Name	R/W	Description
			0: Single conversion mode 1: Continuous conversion mode
2	CAL	R/W	A/D Calibration Calibration starts when this bit is set to 1 by software, and it is cleared by hardware when the calibration is completed. 0: Calibration is completed 1: Start calibration
3	CALRST	R/W	Calibration Reset This bit is set to 1 by software, and is cleared by hardware after the calibration register completes resetting. 0: It means resetting of calibration register is completed 1: Reset calibration register
7:4	Reserved		
8	DMAEN	R/W	DMA Mode Enable 0: Disable 1: Enable Note: Only ADC1 and ADC3 can generate DMA request.
10:9	Reserved		
11	DALIGNCFG	R/W	Data Alignment Mode Configure 0: Right alignment 1: Left alignment
14:12	INJGEXTTRGSEL	R/W	Select the External Trigger Event to Start the Injected Group Conversion Trigger configuration of ADC1 and ADC2 is as follows: 000: TRGO event of timer 1 001: CC4 event of timer 1 010: TRGO event of timer 2 011: CC1 event of timer 2 100: CC4 event of timer 3 101: TRGO event of timer 4 110: EINT Line 15 111：INJSWSC The trigger configuration of ADC3 is as follows: 000: TRGO event of timer 1 001: CC4 event of timer 1 010: CC3 event of timer 4 011: CC2 event of timer 8 100: CC4 event of timer 8 101: TRGO event of timer 5 110: CC4 event of timer 5 111：INJSWSC
15	INJEXTTRGEN	R/W	Enable the External Trigger Conversion Mode of the Injected Channels 0: Disable 1: Enable
16	Reserved		
19:17	REGEXTTRGSEL	R/W	Select the External Trigger Event to Start the Regular Group Conversion

Field	Name	R/W	Description
			Trigger configuration of ADC1 and ADC2 is as follows: 000: CC1 event of timer 1 001: CC2 event of timer 1 010: CC3 event of timer 1 011: CC2 event of timer 2 100: TRGO event of timer 3 110: CC4 event of timer 4 110: EINT Line 11 111：REGSWSC The trigger configuration of ADC3 is as follows: 000: CC1 event of timer 3 001: CC3 event of timer 2 010: CC3 event of timer 1 011: CC1 event of timer 8 100: TRGO event of timer 8 101: CC1 event of timer 5 110: CC3 event of timer 5 111：REGSWSC
20	REGEXTTRGEN	R/W	Enable the External Trigger Conversion Mode of the Regular Channels 0: Disable 1: Enable
21	INJSWSC	R/W	Software Start Conversion Injected Channels If INJSWSC is selected as trigger event in INJEXESEL[2:0] bit, this bit wil be used to start conversion of a group of injected channel; this bit can be set to 1 and cleared by software, and be cleaered by hardware after the conversion is started. 0: Reset state 1: Start conversion of injected channels
22	REGSWSC	R/W	Software Start Conversion Regular Channels If REGSWSC is selected as trigger event in REGEXTSEL[2:0] bit, this bit will be used to start conversion of a group of regular channel; this bit can be set to 1 and cleared by software, and be cleared by hardware after the conversion is started. 0: Reset state 1: Start conversion of regular channels
23	TSVREFEN	R/W	Temperature Sensor And V_{REFINT} Channel Enable This bit is valid only in ADC1. This bit can be set to 1 and cleared by software; in the device with multiple ADCs, this bit only appears in ADC1. 0: Disable 1: Enable
31:24	Reserved		

27.6.4 ADC sampling time register 1 (ADC_SMPTIM1)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
2:0	SMPCYCCFG10	R/W	Channel 10 Sample Cycles Configure 000: 1.5 periods 001: 7.5 periods 010: 13.5 periods 011: 28.5 periods 100: 41.5 periods 101: 55.5 periods 110: 71.5 periods 111: 239.5 periods
5:3	SMPCYCCFG11	R/W	Channel 11\ Sample Cycles Configure Refer to the description of SMPCYCCFG10.
8:6	SMPCYCCFG12	R/W	Channel 12 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
11:9	SMPCYCCFG13	R/W	Channel 13 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
14:12	SMPCYCCFG14	R/W	Channel 14 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
17:15	SMPCYCCFG15	R/W	Channel 15 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
20:18	SMPCYCCFG16	R/W	Channel 16 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
23:21	SMPCYCCFG17	R/W	Channel 17 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
31:24	Reserved		

27.6.5 ADC sampling time register 2 (ADC_SMPTIM2)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
2:0	SMPCYCCFG0	R/W	Channel 0 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
5:3	SMPCYCCFG1	R/W	Channel 1 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
8:6	SMPCYCCFG2	R/W	Channel 2 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
11:9	SMPCYCCFG3	R/W	Channel 3 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
14:12	SMPCYCCFG4	R/W	Channel 4 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
17:15	SMPCYCCFG5	R/W	Channel 5 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
20:18	SMPCYCCFG6	R/W	Channel 6 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
23:21	SMPCYCCFG7	R/W	Channel 7 Sample Cycles Configure

Field	Name	R/W	Description
			Refer to the description of SMPCYCCFG10.
26:24	SMPCYCCFG8	R/W	Channel 8 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
29:27	SMPCYCCFG9	R/W	Channel 9 Sample Cycles Configure Refer to the description of SMPCYCCFG10.
31:30	Reserved		

27.6.6 ADC injected channel data offset register x (ADC_INJDOFx) (x=1..4)

Offset address: 0x14-0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
11:0	INJDOFx	R/W	Data Offset For Injected Channel x When converting the injected channels, these bits define the values to be subtracted from the original converted data, and the result of the conversion can be read in the ADC_INJDATAx register.
31:12	Reserved		

27.6.7 Analog watchdog high-threshold register (ADC_AWDHT)

Offset address: 0x24

Reset value: 0x0000 0FFF

Field	Name	R/W	Description
11:0	AWDHT[11:0]	R/W	Analog Watchdog High Threshold
31:12	Reserved		

27.6.8 Analog watchdog low-threshold register (ADC_AWDLT)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
11:10	AWDLT[11:0]	R/W	Analog Watchdog Low Threshold
31:12	Reserved		

27.6.9 ADC regular sequence register 1 (ADC_REGSEQ1)

Offset address: 0x2C

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	REGSEQC13	R/W	13th Conversion In Regular Sequence Define the channel number of No. 13 conversion in regular sequence (0~17)
9:5	REGSEQC14	R/W	14th Conversion In Regular Sequence Refer to the description of REGSEQC13.

Field	Name	R/W	Description
14:10	REGSEQC15	R/W	15th Conversion In Regular Sequence Refer to the description of REGSEQC13.
19:15	REGSEQC16	R/W	16th Conversion In Regular Sequence Refer to the description of REGSEQC13.
23:20	REGSEQLEN	R/W	Regular Channel Sequence Length These bits are defined by software as the number of channels in regular channel conversion sequence. 0000: One conversion 0001: Two conversions …… 1111: 16 conversions
31:24	Reserved		

27.6.10 ADC regular sequence register 2 (ADC_REGSEQ2)

Offset address: 0x30

Reset value: 0x0000 000

Field	Name	R/W	Description
4:0	REGSEQC7	R/W	7th Conversion In Regular Sequence Refer to the description of REGSEQC13.
9:5	REGSEQC8	R/W	8th Conversion In Regular Sequence Refer to the description of REGSEQC13.
14:10	REGSEQC9	R/W	9th Conversion In Regular Sequence Refer to the description of REGSEQC13.
19:15	REGSEQC10	R/W	10th Conversion In Regular Sequence Refer to the description of REGSEQC13.
24:20	REGSEQC11	R/W	11th Conversion In Regular Sequence Refer to the description of REGSEQC13.
29:25	REGSEQC12	R/W	12th Conversion In Regular Sequence Refer to the description of REGSEQC13.
31:30	Reserved		

27.6.11 ADC regular sequence register 3 (ADC_REGSEQ3)

Offset address: 0x34

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	REGSEQC1	R/W	1st Conversion In Regular Sequence Refer to the description of REGSEQC13.
9:5	REGSEQC2	R/W	2nd Conversion In Regular Sequence Refer to the description of REGSEQC13.
14:10	REGSEQC3	R/W	3rd Conversion In Regular Sequence Refer to the description of REGSEQC13.
19:15	REGSEQC4	R/W	4th Conversion In Regular Sequence Refer to the description of REGSEQC13.

Field	Name	R/W	Description
24:20	REGSEQC5	R/W	5th Conversion In Regular Sequence Refer to the description of REGSEQC13.
29:25	REGSEQC6	R/W	6th Conversion In Regular Sequence Refer to the description of REGSEQC13.
31:30	Reserved		

27.6.12 ADC injected sequence register (ADC_INJSEQ)

Offset address: 0x38

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	INJSEQC1	R/W	1st Conversion In Injected Sequence Define the channel number of No. 1 conversion in injected sequence (0~17)
9:5	INJSEQC2	R/W	2nd Conversion In Injected Sequence
14:10	INJSEQC3	R/W	3rd Conversion In Injected Sequence
19:15	INJSEQC4	R/W	4th Conversion In Injected Sequence
21:20	INJSEQLEN	R/W	Injected Channel Sequence Length These bits are defined by software as the number of channels in injected channel conversion sequence, and the conversion sequence is: $\text{INJSEQC}_{(4\text{-INJSEQLEN})}$ →$\text{INJSEQ}_{(5\text{-INJSEQLEN})}$ →$\text{INJSEQC}_{(6\text{-INJSEQLEN})}$ →$\text{INJSEQC}_{(7\text{-INJSEQLEN})}$; the details are as follows: 00: One conversion, only converting INJSEQC4 01: Two conversions; the conversion sequence is INJSEQC3→INJSEQC4 10: Three conversions; the conversion sequence is INJSEQC2→INJSEQC3→INJSEQC4 11: Four conversions; the conversion sequence is INJSEQC1→INJSEQC2→INJSEQC3→INJSEQC4
31:22	Reserved		

27.6.13 ADC injected data register x (ADC_INJDATAx) (x= 1..4)

Offset address: 0x3C–0x48

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	INJDATA	R	Injected Conversion Data Conversion result of injected channel, read-only.
31:16	Reserved		

27.6.14 ADC regular data register (ADC_REGDATA)

Offset address: 0x4C

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	REGDATA	R	Regular Conversion Data Conversion result of regular channel, read-only.
31:16	ADC2DATA	R	ADC2 Conversion Data (1) This bit is valid in ADC1, indicating the result of ADC2 regular channel conversion in dual-ADC mode; (2) This bit is reserved in ADC2 and ADC3.

28 Digital-to-analog Converter (DAC)

28.1 Full Name and Abbreviation Description of Terms

Table 143 Full Name and Abbreviation Description of DAC Terms

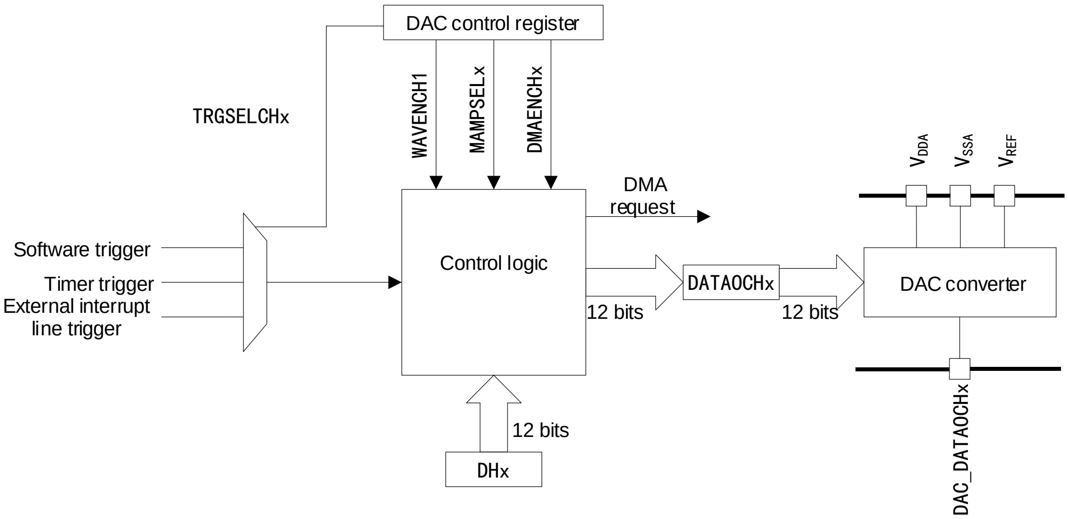
Full name in English	English abbreviation
Linear Feedback Shift Register	LFSR

28.2 Introduction

DAC is a digital/analog converter that can be configured to input 8-bit or 12-bit data and output voltage. When 12-bit data is input, the data can be set to be left aligned or right aligned. It has two-way DAC output channels, which do not affect each other; each channel has multiple trigger sources to trigger conversion; a single channel can trigger conversion output, or both channels can trigger conversion output at the same time. Both channels can generate noise waveform and triangle waveform independently.

28.3 Structure Block Diagram

Figure 133 DAC Structure Block Diagram



28.4 Functional Description

28.4.1 DAC Conversion and Trigger Source

DAC output can obtain corresponding voltage value by calculating the data in DAC_DATAOCHx register. However, it is impossible to write data directly to DAC_DATAOCHx register, and it is required to write to DAC_DHx register and then through corresponding trigger, load the data in DAC_DHx to

DAC_DATAOCHx.

When the channel trigger is disabled (TRGENCHx bit in the register DAC_CTRL is set to 0), write the value in DAC_DHx register and it will be automatically transferred to DAC_DATAOCHx after one APB1 clock cycle.

When the channel trigger is enabled (TRGENCHx bit in the register DAC_CTRL is set to 1), write the value in DAC_DHx register and it will be transferred to DAC_DATAOCHx after different clock cycles according to the selected trigger source. Three types of trigger sources can be selected:

- Timer update event
- External interrupt trigger
- Software trigger

When the update event and external interrupt of the timer are selected as the trigger source, the transmission will be completed after three APB clock cycles; when software trigger is selected, the transmission will be completed after one APB1 clock cycle.

When transmitting the data to DAC_DATAOCHx register, after a period of time, the digital quantity will be outputted after it is converted linearly into analog voltage. The intermediate conversion time will vary according to the power supply voltage and the analog output load.

28.4.2 DAC Reference Voltage and Output

DAC uses V_{REF} as reference voltage and by grounding the V_{SSA}, the output voltage range of DAC can be obtained, namely: 0-V_{REF}.

DAC output calculation formula is: DAC output=V_{REF} * (DATAOCHx/4095).

28.4.3 DAC Data Format

Single-channel DAC

The registers that are written in three modes are as follows

- 8-bit data right aligned: DAC_DH8Rx[7:0]
- 12-bit data left aligned: DAC_DH12Lx[15:4]
- 12-bit data right aligned: DAC_DH12Rx[11:0]

Double-channel DAC

The registers that are written in three modes are as follows

- 8-bit data right aligned: DAC_DH8RD[15:0]
- 12-bit data left aligned: DAC_DH12LDUAL[15:4], DAC_DH12LDUAL[31:20]
- 12-bit data right aligned: DAC_DH12RDUAL[11:0], DAC_DH12RDUAL[27:16]

28.4.4 DAC Waveform Generation

Each channel of DAC can independently generate noise and triangle wave.

28.4.5 DAC Double-channel Conversion

When two channels work at the same time, the written data can be written to the common registers: DH8RDUAL, DH12RDUAL and DH12LDUAL, so as to effectively use the bus bandwidth.

Dual-channel conversion can be divided into independent conversion and synchronous conversion. The specific configuration and description are as follows.

28.4.5.1 Independent trigger

Waveform generator disabled

（1） Enable two-channel trigger mode;

（2） Configure two channels and use different trigger sources.

Use the same LFSR

（1） Enable two-channel trigger mode;

（2） Configure two channels and use different trigger sources;

（3） Enable the noise generation function of two channels, and set the same LFSR mask value.

Use different LFSR

（1） Enable two-channel trigger mode;

（2） Configure two channels and use different trigger sources;

（3） Enable the noise generation function of two channels, and set different LFSR mask value.

Generate the same triangle wave

（1） Enable two-channel trigger mode;

（2） Configure two channels and use different trigger sources;

（3） Enable the triangle wave generation function of two channels, and set the same triangular amplitude.

Generate different triangle wave

（1） Enable two-channel trigger mode;

（2） Configure two channels and use different trigger sources;

（3） Enable the triangle wave generation function of two channels, and set different triangular amplitude.

28.4.5.2 Synchronous trigger

Synchronous software startup

Disable the trigger mode of two channels; after writing data, wait for one APB1 clock cycle and then transfer to DAC_DATAOCH1 and DAC_DATAOCH2 registers at the same time.

Waveform generator disabled

（1） Enable two-channel trigger mode;

（2） Configure two channels and use the same trigger source.

Use the same LFSR

（1） Enable two-channel trigger mode;

（2） Configure two channels and use the same trigger source;

（3） Enable the noise generation function of two channels, and set the same LFSR mask value.

Use different LFSR

（1） Enable two-channel trigger mode;

（2） Configure two channels and use the same trigger source;

（3） Enable the noise generation function of two channels, and set different LFSR mask value.

Generate the same triangle wave

（1） Enable two-channel trigger mode;

（2） Configure two channels and use the same trigger source;

（3） Enable the triangle wave generation function of two channels, and set the same triangular amplitude.

Use different triangle wave

（1） Enable two-channel trigger mode;

（2） Configure two channels and use the same trigger source;

（3） Enable the triangle wave generation function of two channels, and set different triangular amplitude.

28.5 Register Address Mapping

Table 144 DAC Register Address Mapping

Register name	Description	Offset address
DAC_CTRL	DAC control register	0x00
DAC_SWTRG	DAC software trigger register	0x04
DAC_DH12R1	DAC Channel 1 12-bit right-aligned data holding register	0x08
DAC_DH12L1	DAC Channel 1 12-bit left-aligned data holding register	0x0C
DAC_DH8R1	DAC Channel 1 8-bit right-aligned data holding register	0x10
DAC_DH12R2	DAC Channel 2 12-bit right-aligned data holding register	0x14
DAC_DH12L2	DAC Channel 2 12-bit left-aligned data holding register	0x18
DAC_DH8R2	DAC Channel 2 8-bit right-aligned data holding register	0x1C
DAC_DH12RDUAL	Dual-DAC 12-bit right-aligned data holding register	0x20
DAC_DH12LDUAL	Dual-DAC 12-bit left-aligned data holding register	0x24
DAC_DH8RDUAL	Dual-DAC 8-bit right-aligned data holding register	0x28
DAC_DATAOCH1	DAC Channel 1 data output register	0x2C
DAC_DATAOCH2	DAC Channel 2 data output register	0x30

28.6 Register Functional Description

28.6.1 DAC control register (DAC_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ENCH1	R/W	DAC Channel1 Enable 0: Disable 1: Enable
1	BUFFDCH1	R/W	DAC Channel1 Output Buffer Disable 0: Enable 1: Disable
2	TRGENCH1	R/W	DAC Channel1 Trigger Enable 0: Disable 1: Enable
5:3	TRGSELCH1	R/W	DAC Channel1 Trigger Source Select The trigger source can be selected through this register when Channel 1 trigger is enabled (TRGENCH1=1).

Field	Name	R/W	Description
			000: TMR6 TRGO event 001: TMR8 TRGO event 010: TMR7 TRGO event 011: TMR5 TRGO event 100: TMR2 TRGO event 101: TMR4 TRGO event 110: External interrupt line 9 111: Software trigger
7:6	WAVENCH1	R/W	DAC Channel1 Noise/Triangle Wave Generation Enable 00: Waveform is not generated 10: Noise waveform is generated 1x: Triangle waveform is generated
11:8	MAMPSELCH1	R/W	Select DAC Channel1 LFSR Bit Mask/Triangle Wave Amplitude Selector In the mode of "generating LFSR noise", select the bit to mask LFSR through this bit; In the mode of "generating triangle wave", select the amplitude of triangle wave through this bit. 0000: Unmask LFSR bit [0]/triangle wave amplitude is 1 0001: Unmask LFSR bit [1:0]/triangle wave amplitude is 3 0010: Unmask LFSR bit [2:0]/triangle wave amplitude is 7 0011: Unmask LFSR bit [3:0]/triangle wave amplitude is 15 0100: Unmask LFSR bit [4:0]/triangle wave amplitude is 31 0101: Unmask LFSR bit [5:0]/triangle wave amplitude is 63 0110: Unmask LFSR bit [6:0]/triangle wave amplitude is 127 0111: Unmask LFSR bit [7:0]/triangle wave amplitude is 255 1000: Unmask LFSR bit [8:0]/triangle wave amplitude is 511 1001: Unmask LFSR bit [9:0]/triangle wave amplitude is 1023 1010: Unmask LFSR bit [10:0]/triangle wave amplitude is 2047 ≥1011: Unmask LFSR bit [11:0] / triangle wave amplitude is 4095
12	DMAENCH1	R/W	DAC Channel1 DMA Enable 0: Disable 1: Enable
15:13	Reserved		
16	ENCH2	R/W	DAC Channel2 Enable 0: Disable 1: Enable
17	BUFFDCH2	R/W	DAC Channel2 Output Buffer Disable 0: Enable 1: Disable
18	TRGENCH2	R/W	DAC Channel2 Trigger Enable 0: Disable 1: Enable
21:19	TRGSELCH2	R/W	DAC Channel2 Trigger Source Select The trigger source can be selected through this register when Channel 2 trigger is enabled (TRGENCH2=1)

Field	Name	R/W	Description
			000: TMR6 TRGO event 001: TMR8 TRGO event 010: TMR7 TRGO event 011: TMR5 TRGO event 100: TMR2 TRGO event 101: TMR4 TRGO event 110: External interrupt line 9 111: Software trigger
23:22	WAVENCH2	R/W	DAC Channel2 Noise/Triangle Wave Generation Enable 00: Waveform is not generated 01: Noise waveform is generated 1x: Triangle waveform is generated
27:24	MAMPSELCH2	R/W	Select DAC Channel2 LFSR Bit Mask/Triangle Wave Amplitude Selector In the mode of "generating LFSR noise", select the bit to mask LFSR through this bit; In the mode of "generating triangle wave", select the amplitude of triangle wave through this bit. 0000: Unmask LFSR bit [0]/triangle wave amplitude is 1 0001: Unmask LFSR bit [1:0]/triangle wave amplitude is 3 0010: Unmask LFSR bit [2:0]/triangle wave amplitude is 7 0011: Unmask LFSR bit [3:0]/triangle wave amplitude is 15 0100: Unmask LFSR bit [4:0]/triangle wave amplitude is 31 0101: Unmask LFSR bit [5:0]/triangle wave amplitude is 63 0110: Unmask LFSR bit [6:0]/triangle wave amplitude is 127 0111: Unmask LFSR bit [7:0]/triangle wave amplitude is 255 1000: Unmask LFSR bit [8:0]/triangle wave amplitude is 511 1001: Unmask LFSR bit [9:0]/triangle wave amplitude is 1023 1010: Unmask LFSR bit [10:0]/triangle wave amplitude is 2047 ≥1011: Unmask LFSR bit [11:0] / triangle wave amplitude is 4095
28	DMAENCH2	R/W	DAC Channel2 DMA Enable 0: Disable 1: Enable
31:29	Reserved		

28.6.2 DAC software trigger register (DAC_SWTRG)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	SWTRG1	W	DAC Channel1 Software Trigger Enable This bit can be set to 1 and cleared by software; once the data in the register DAC_DH1 is transferred to the register DAC_DATAOCH1, this bit will be cleared by hardware. 0: Disable 1: Enable

Field	Name	R/W	Description
1	SWTRG2	W	DAC Channel2 Software Trigger Enable This bit can be set to 1 and cleared by software; once the data in the register DAC_DH2 is transferred to the register DAC_DATAOCH2, this bit will be cleared by hardware. 0: Disable 1: Enable
31:2	Reserved		

28.6.3 DAC Channel 1 12-bit right-aligned data holding register (DAC_DH12R1)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
11:0	DATA	R/W	DAC Channel1 12-bit Right-Aligned Data This bit is written by the software, representing the data of 12-bit DAC channel 1.
31:12	Reserved		

28.6.4 DAC Channel 1 12-bit left-aligned data holding register (DAC_DH12L1)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	Reserved		
15:4	DATA	R/W	DAC Channel1 12-Bit Left-Aligned Data
31:16	Reserved		

28.6.5 DAC Channel 1 8-bit right-aligned data holding register (DAC_DH8R1)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	DATA	R/W	DAC Channel1 8-bit Right-Aligned Data
31:8	Reserved		

28.6.6 DAC Channel 2 12-bit right-aligned data holding register (DAC_DH12R2)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
11:0	DATA	R/W	DAC Channel2 12-bit Right-Aligned Data

Field	Name	R/W	Description
31:12	Reserved		

28.6.7 DAC Channel 2 12-bit left-aligned data holding register (DAC_DH12L2)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	Reserved		
15:4	DATA	R/W	DAC Channel2 12-Bit Left-Aligned Data
31:16	Reserved		

28.6.8 DAC Channel 2 8-bit right-aligned data holding register (DAC_DH8R2)

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	DATA	R/W	DAC Channel2 8-bit Right-Aligned Data
31:8	Reserved		

28.6.9 Dual-DAC 12-bit right-aligned data holding register (DAC_DH12RDUAL)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
11:0	DATACH1	R/W	DAC Channel1 12-bit Right-Aligned Data
15:12	Reserved		
27:16	DATACH2	R/W	DAC Channel2 12-bit Right-Aligned Data
31:28	Reserved		

28.6.10 Dual-DAC 12-bit left-aligned data holding register (DAC_DH12LDUAL)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	Reserved		
15:4	DATACH1	R/W	DAC Channel1 12-Bit Left-Aligned Data
19:16	Reserved		
31:20	DATACH2	R/W	DAC Channel212-Bit Left-Aligned Data

28.6.11 Dual-DAC 8-bit right-aligned data holding register (DAC_DH8RDUAL)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	DATACH1	R/W	DAC Channel1 8-bit Right-Aligned Data
15:8	DATACH2	R/W	DAC Channel2 8-bit Right-Aligned Data
31:16	Reserved		

28.6.12 DAC Channel 1 data output register (DAC_DATAOCH1)

Offset address: 0x2C

Reset value: 0x0000 0000

Field	Name	R/W	Description
11:0	DATA	R	DAC Channel1 Data Output
31:12	Reserved		

28.6.13 DAC Channel 2 data output register (DAC_DATAOCH2)

Offset address: 0x30

Reset value: 0x0000 0000

Field	Name	R/W	Description
11:0	DATA	R	DAC Channel2 Data Output
31:12	Reserved		

29 Cyclic Redundancy Check Computing Unit (CRC)

29.1 Introduction

The Cyclic Redundancy Check (CRC) computing unit can get 32-bit CRC computing result by calculating the input data through a fixed generator polynomial, which is mainly used to detect or verify the correctness and integrity of the data after transmission or saving.

29.2 Functional Description

29.2.1 Calculation Method

Use CRC-32 (Ethernet) polynomial: 0x4C11DB7

$$（X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1）$$

29.2.2 Calculation Time

The calculation time is four AHB clock cycles.

Every time a new data is written, the result will be a combination of the last calculation result and the new calculation result. (Execute operation for the whole word). Write operation of CPU will be suspended during calculation, so that "Back-to-back" write or continuous "read" -"write" operation can be performed for the register CRC_DATA.

29.3 Register Address Mapping

Table 145 CRC Computing Unit Register Address Mapping

Register name	Description	Offset address
DATA	Data register	0x00
INDATA	Independent data register	0x04
CTRL	Control register	0x08

29.4 Register Functional Description

CRC computing unit contains two data memories and one control memory.

29.4.1 Data register (CRC_DATA)

Offset address: 0x00

Reset value: 0xFFFF FFFF

Field	Name	R/W	Description
31:0	DATA	R/W	32bit Data Register It is used as a new data input port for CRC computing during write operation; when the read operation is executed, the result of CRC computing is returned.

29.4.2 Independent data register (CRC_INDATA)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	INDATA	R/W	Independent 8bit Data Register It can be used to temporarily store 1-byte data; the CRC reset generated by RST bit of the register CRC_CTRL has no effect on this register.
31:8	Reserved.		

Note: This register does not take part in calculation and can store any data.

29.4.3 Control register (CRC_CTRL)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	RST	W	Reset CRC Calculation Unit After reset, set the data register as 0xFFFF FFFF, and this bit can only be written as 1 by software and automatically cleared by hardware.
31:1	Reserved		

30 Floating-point Unit (FPU)

30.1 Introduction

FPU (floating point unit) fully supports operations of single precision addition, subtraction, multiplication, division, multiplication, accumulation and square root. It also provides conversion between fixed-point and floating-point data formats, and floating-point constant instructions. FPU provides floating-point estimation function, which conforms to ANSI (American National Standards Institute)/IEEE (Institute of Electrical and Electronics Engineers) 754-2008 standard and IEEE binary floating-point algorithm standard, which is called IEEE 754 standard.

Note: Not apply to APM32E103VET6S.

30.2 Functional Description

30.2.1 Supported Algorithms

FPU only supports single precision and supports IEEE754 compliance, including algorithms of: CMP, SUM, SUB, PRDCT, MAC, DIV, INVRGSQT, RGSQT, SUMSQ, DOT, conversion from floating point to integer and conversion from integer to floating point.

Table 146 FPU Supported Algorithm

Term	Definition
I2FP	Conversion from integer to floating point
FP2I	Conversion from floating point to integer
CMP（p, q）	Comparison of p and q
SUM（p, n）	$SUM(p,n) = \sum_{（i=0,n）} pi = p0 + p1 + .. + pi + .. + pn$, 0≤n<16
SUB（p, n）	$SUB(p,n) = p0 - \sum_{（i=0,n）} pi = p0 - p1 - .. - pi - .. - pn$, 0≤n<16
PRDCT（p, n）	Product：$PRDCT(p.n) = \prod_{（i=0,n）} pi = p0 * p1 * .. * pi * .. * pn$, 0≤n<16
DIV（p, q）	Division:$DIV(p,q) = \frac{p}{q}$
INVSQRT（p）	Square Root：INVSQRT（p）=$\frac{1}{\sqrt{p}}$
MAC（p, q, s）	Floating-Point Multiply-Add：MAC（p, q, s） $= p * q + s$
SUMSQ（p, n）	∑ (i=1, n) pi2, wherein p is the vector of the length n, 0≤n<16
DOT（p, q, n）	∑ (i=1, n) (pi * qi), whrein p and q are the vector of the length n, 0≤n<16
SIN（x）	$x \in (-\infty, +\infty)$, wherein x is expressed in radian
COS（x）	$x \in (-\infty, +\infty)$, wherein x is expressed in radian
TAN（x）	$x \in (-\infty, +\infty)$, wherein x is expressed in radian

Term	Definition
ASIN（x）	$x \in [-1,+1]$, wherein the output value is expressed in radian, and is in the interval of$[-\pi/2,+\pi/2]$
ACOS（x）	$x \in [-1,+1]$, wherein the output value is expressed in radian, and is in the interval of $[0,\pi]$
ATAN（x）	$x \in [-\infty,+\infty]$, wherein the output value is expressed in radian, and is in the interval of $[-\pi/2,+\pi/2]$
ATAN2（y, x）	$x, y \in [-\infty,+\infty]$, wherein the output value is expressed in radian, and is in the interval of $[-\pi,+\pi]$

30.2.2 Software Programming

30.2.2.1 Programming process

It is recommended to use the relocatable library provided by us to connect the user program instead of programming on the basis of registers.

30.2.2.2 Single-operand operation

For FP2I, I2FP, INVSQRT, SIN, COS, TAN, ASIN, ACOS and ATAN, the recommended configuration is as follows:

（1） Configure CTRL register. Select the required operation mode and set the start (STR) bit. For INVSQRT mode, set the cycle index to improve the accuracy of CTRL[7:6].

（2） Write the operand to FPU_FPI1

（3） Flag of waiting for interrupt done or checking the FPU_ISTS register done (DONE)

（4） Read the result from FPU_FPO

30.2.2.3 Operation of two operands

For DIV, the recommended configuration is as follows:

（1） Configure FPU_CTRL register, select DIV mode and set the start (STR) bit.

（2） Write the first operand to FPU_FPI1

（3） Write the second operand to FPU_FPI2

（4） Flag of waiting for interrupt done or checking the FPU_ISTS register done (DONE)

（5） Read the result from FPU_FPO

For CMP, the recommended configuration is as follows:

（1） Configure FPU_CTRL register, select CMP mode and set the start (STR) bit.

（2） Write the first operand to FPU_FPI1

（3） Write the second operand to FPU_FPI2

（4） Flag of waiting for interrupt done or checking the FPU_ISTS register done (DONE)

（5） Read the CMP result from FPU_ISTS[3:2]

For ATAN2, the recommended configuration is as follows:

（1） Configure FPU_CTRL register, select ATAN2 mode and set the start (STR) bit.

（2） Write the first operand to FPU_FPI1

（3） Write the second operand to FPU_FPI2

（4） Flag of waiting for interrupt done or checking the FPU_ISTS register done (DONE)

（5） Read the result from FPU_FPO

30.2.2.4 Operation of three operands

For MAC, the recommended configuration is as follows:

（1） Configure FPU_CTRL register, select MAC mode and set the start (STR) bit.

（2） Write the first operand to FPU_FPI1

（3） Write the second operand to FPU_FPI2

（4） Write the third operand to FPU_FPI2

（5） Flag of waiting for interrupt done or checking the FPU_ISTS register done (DONE)

（6） Read the result from FPU_FPO

30.2.2.5 Operation of n operands

For SUM, SUB, PRDCT and SUMSQ, the recommended configuration is as follows:

（1） Configure FPU_CTRL register, select the required mode, enter the value of n in the nLEN field and set the start (STR) bit.

（2） Write the P0 operand to FPU_FPI1

（3） Write the operand of P1..n-1 to FPU_FPI2

（4） Flag of waiting for interrupt done or checking the FPU_ISTS register done (DONE)

（5） Read the result from FPU_FPO

For DOT, the recommended configuration is as follows:

（1） Configure FPU_CTRL register, select DOT mode, enter the value of n in the nLEN field and set the start (STR) bit.

（2） Write the P0 operand to FPU_FPI1

（3） Write the Q0 operand to FPU_FPI2

（4） Write the P1 operand to FPU_FPI1

（5） Write the Q1 operand to FPU_FPI2

（6） …

（7） Write the Pn-1 operand to FPU_FPI1

（8） Write the Qn-1 operand to FPU_FPI2

（9） Flag of waiting for interrupt done or checking the FPU_ISTS register done (DONE)

（10） Read the result from FPU_FPO

30.3 Register Address Mapping

Table 147 FPU Register Address Mapping

Register name	Description	Offset address
FPU_CTRL	Control register	0x00
FPU_ISTS	Interrupt state register	0x04
FPU_FPO	Floating-point output value register	0x08
FPU_FPI1	Floating-point input value register 0	0x0C
FPU_FPI2	Floating-point input value register 1	0x10
FPU_IM	Interrupt mask register	0x14

30.4 Register Functional Description

30.4.1 Control register (FPU_CTRL)

Offset address: 0x0

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	STR	R/W	Start FPU 0: No effect 1: Start FPU. This bit can be automatically cleared by hardware after startup.
5:1	OP	R/W	FPU operation mode 0：ASIN 1：ACOS 2：ATAN 3：ATAN2 4：SIN 5：COS 6：TAN 7~15：Reserved 16：SUM（p, n）, Summation 17：SUMSQ（p, n）, Sum of squares 18：SUB（p, n）, Subtraction 19：PRDCT（p, n）, Production 20：DOT（p, q, n）, Dot Production 21：MAC（p, q, s）, Floating-Point Multiply-Add 22：DIV（p, q）, Division 23：INVSQRT（p）, Square Root 24：CMP（p,q） 25: I2FP, conversion from integer to floating point 26: FP2I, conversion from floating point to integer 27~31: Reserved
7:6	ITERTION	R/W	Improve the repetition time of accuracy algorithm
11:8	nLEN	R/W	The length of the equation SUM (p, n), SUB (p, n), SUMSQR (p, n), PRDCT (p, n), and DOT (p, q, n).
31:12	Reserved		

30.4.2 Interrupt state register (FPU_ISTS)

Offset address: 0x4

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	DONE	R/W	Flag of FPU calculation completion 0: FPU calculation uncompleted 1: FPU calculation completed
1	BUSY	R/W	FPU busy flag 0: FPU in idle state 1: FPU in busy state This bit can be set by hardware after FPU runs, and automatically cleared by hardware after FPU completes calculation.
3:2	CMPR[3:2]	R/W	Comparison algorithm result register 0：a = b. 1：a < b. 2：a > b.

Field	Name	R/W	Description
			3: When one input value is not a number (NaN), it is unordered
4	ZERO	R/W	Zero flag 0: Meaningless. 1: The calculation result is zero or is zero after round-off
5	INF	R/W	Infinitely great 0: Meaningless 1: The calculation result is infinitely great or is infinitely great after round-off
6	INVALID	R/W	Floating-point operation is invalid 0: Meaningless 1: Illegal input operation
7	TINY	R/W	The calculation result is tiny 0: Meaningless. 1: After round-off, the floating point is a quantity less than the minimum normalized number, but not an exact zero.
8	HUGE	R/W	The calculation result is great. 0: Meaningless. 1: After round-off, the finite floating point result is a quantity greater than the maximum normalized number.
9	INEXACT	R/W	The calculation result is not accurate 0: Meaningless. 1: The integer or floating point output value is not equal to infinitely great accurate result
10	HUGEINT	R/W	Great integer complement 0: Meaningless 1: After round-off, the integer result is a quantity greater than the maximum representable number of complement of two identical symbols
11	DIVBY0	R/W	Division by zero 0: Meaningless 1: Run the operation of division by zero
31:12	Reserved		

30.4.3 Floating-point output value register (FPU_FPO)

Offset address: 0x8

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	FPO	R	Floating-point output value register

30.4.4 Floating-point input value register 1 (FPU_FPI1)

Offset address: 0xC

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	FPI1	R/W	Floating-point input value register 0

30.4.5 Floating-point input value register 2 (FPU_FPI2)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	FPI2	R	Floating-point input value register 1

30.4.6 Interrupt mask register (FPU_IM)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	DONE	R/W	Operation completion interrupt mask 0: Operation completion interrupt enabled 1: Operation completion interrupt masked
1	INVALID	R/W	Operation interrupt mask invalid 0: Invalid operation interrupt enabled 1: Invalid operation interrupt masked
31:2	Reserved		

31 Chip electronic signature

The chip electronic signature includes flash capacity information of main memory and 96-bit unique chip ID, which have been written into the system memory area of the chip before leaving the factory, and are read-only and can not be modified by users.

31.1 Capacity register of main memory area

31.1.1 Flash capacity register (16 bits)

Base address: 0x1FFF F7E0

Read-only, the value has been prepared before leaving the factory

Field	Name	R/W	Description
15:0	F_SIZE	R	Flash Size Indicate the capacity of main memory area of the product (in KB). For example: 0x0080=128 KB

31.2 96-bit unique chip ID

Purposes of unique ID may be:

- As serial number (such as USBD character serial number or other terminal application)
- As a password; this unique identification can be used with software encryption and decryption algorithm to improve the security of the code in flash memory when writing the flash memory
- Used to activate the startup process with security mechanism
- The reference number provided by the identity is unique to any MCU series. Users cannot change the unique ID under no circumstances. According to different usage, users can choose to read the identity in byte, half word, or full word.

Base address: 0x1FFF F7E8

Offset address: 0x00

Read-only, the value has been prepared before leaving the factory

Field	Name	R/W	Description
15:0	U_ID[15:0]	R	Unique identity flag 15:0 bits

Offset address: 0x02

Read-only, the value has been prepared before leaving the factory

Field	Name	R/W	Description
15:0	U_ID[31:16]	R	Unique identity flag 31:16 bits

Offset address: 0x04

Read-only, the value has been prepared before leaving the factory

Field	Name	R/W	Description
31:0	U_ID[63:32]	R	Unique identity flag 63:32 bits

Offset address: 0x08

Read-only, the value has been prepared before leaving the factory

Field	Name	R/W	Description
31:0	U_ID[95:64]	R	Unique identity flag 95:64 bits

32 Version History

Table 148 Document Version History

Date	Version	Change History
August 6, 2021	1.0	New
March 15, 2022	1.1	(1) Modify the SDIO module (2) Modify the DMC_CTRL2 register (3) Modify the contents of the CAN communication mode section (4) Modify WDTSEL bit of Option Bytes in Flash module. (5) Modify the filter register of the CAN module (6) "SCLKSW" change to "SCLKSEL", "SCLKSWSTS" change to "SCLKSELSTS"
June 21,2022	1.2	(1) Modify the DMC module configuration examples and offset addresses and main characteristics (2) Modify the SMC module timing parameters and offset addresses (3) Modify Arm trademark (4) Add the statement

Statement

This document is formulated and published by Geehy Semiconductor Co., Ltd. (hereinafter referred to as "Geehy"). The contents in this document are protected by laws and regulations of trademark, copyright and software copyright. Geehy reserves the right to make corrections and modifications to this document at any time. Please read this document carefully before using Geehy products. Once you use the Geehy product, it means that you (hereinafter referred to as the "users") have known and accepted all the contents of this document. Users shall use the Geehy product in accordance with relevant laws and regulations and the requirements of this document.

1. Ownership

This document can only be used in connection with the corresponding chip products or software products provided by Geehy. Without the prior permission of Geehy, no unit or individual may copy, transcribe, modify, edit or disseminate all or part of the contents of this document for any reason or in any form.

The "极海" or "Geehy" words or graphics with "®" or "TM" in this document are trademarks of Geehy. Other product or service names displayed on Geehy products are the property of their respective owners.

2. No Intellectual Property License

Geehy owns all rights, ownership and intellectual property rights involved in this document.

Geehy shall not be deemed to grant the license or right of any intellectual property to users explicitly or implicitly due to the sale or distribution of Geehy products or this document.

If any third party's products, services or intellectual property are involved in this document, it shall not be deemed that Geehy authorizes users to use the aforesaid third party's products, services or intellectual property, unless otherwise agreed in sales order or sales contract.

3. Version Update

Users can obtain the latest document of the corresponding models when ordering Geehy products.

If the contents in this document are inconsistent with Geehy products, the agreement in thesales order or the sales contract shall prevail.

4. Information Reliability

The relevant data in this document are obtained from batch test by Geehy Laboratory or cooperative third-party testing organization. However, clerical errors in correction or errors caused by differences in testing environment may occur inevitably. Therefore, users should understand that Geehy does not bear any responsibility for such errors that may occur in this document. The relevant data in this document are only used to guide users as performance parameter reference and do not constitute Geehy’s guarantee for any product performance.

Users shall select appropriate Geehy products according to their own needs, and effectively verify and test the applicability of Geehy products to confirm that Geehy products meet their own needs, corresponding standards, safety or other reliability requirements. If loses are caused to users due to the user’s failure to fully verify and test Geehy products, Geehy will not bear any responsibility.

5. Legality

USERS SHALL ABIDE BY ALL APPLICABLE LOCAL LAWS AND REGULATIONS WHEN USING THIS DOCUMENT AND THE MATCHING GEEHY PRODUCTS. USERS SHALL UNDERSTAND THAT THE PRODUCTS MAY BE RESTRICTED BY THE EXPORT, RE-EXPORT OR OTHER LAWS OF THE COUNTIRIES OF THE PRODUCTS SUPPLIERS, GEEHY, GEEHY DISTRIBUTORS AND USERS. USERS (ON BEHALF OR ITSELF, SUBSIDIARIES AND AFFILIATED ENTERPRISES) SHALL AGREE AND PROMISE TO ABIDE BY ALL APPLICABLE LAWS AND REGULATIONS ON THE EXPORT AND RE-EXPORT OF GEEHY PRODUCTS AND/OR TECHNOLOGIES AND DIRECT PRODUCTS.

6. Disclaimer of Warranty

THIS DOCUMENT IS PROVIDED BY GEEHY "AS IS" AND THERE IS NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

GEEHY WILL BEAR NO RESPONSIBILITY FOR ANY DISPUTES ARISING FROM THE SUBSEQUENT DESIGN OR USE BY USERS.

7. Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL GEEHY OR ANY OTHER PARTY WHO PROVIDE THE DOCUMENT "AS IS", BE LIABLE FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE DOCUMENT (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY USERS OR THIRD PARTIES).

8. Scope of Application

The information in this document replaces the information provided in all previous versions of the document.