User Manual

G32A1085

G32A1065

G32A1045

Arm® Cortex®-M0+ based 32-bit

Automotive-grade MCU

Version: V1.2

Contents

1 Introduction and Document Description Rules

1.1 Introduction

This reference manual provides all the information about how to use MCU (Microcontroller Unit) system architecture, memory and peripherals.

For information about Arm® Cortex®-M0+ core, refer to Arm® Cortex®-M0+ technical reference manual; refer to the corresponding datasheet for detailed data such as model information, dimensions and electrical characteristics of the device; for all MCU series models, refer to the corresponding data manual for memory mapping, peripheral existence and their number.

Note: Zhuhai Geehy Semiconductor Co., Ltd. is hereinafter referred to as "Geehy".

1.2 Document Description Rules

1.2.1 "Register Functional Description" Rules

(1) Control (CTRL) registers are all "set to 1 and cleared to 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 status 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 R/W Abbreviation and Description

R/W	Description	Abbreviation
read/write	The software can read and write this bit.	R/W
read-only	The software can only read this bit.	R
write-only	The 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	The software can read this bit and clear it by writing data.	RC_W

R/W	Description	Abbreviation
read/clear by read	The software can read this bit, reading this bit will automatically clear it to 0, and writing this bit is invalid.	RC_R
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 reverse this bit only by writing 1, and writing 0 has no effect on this bit.	T

Table 2 Functional Description and Full Names and Abbreviations of Terms of Commonly Used Registers

Full Name	Abbreviation
Enable	EN
Disable	D
Clear	CLR
Select	SEL
Configure	CFG
Control	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

Full Name	Abbreviation
Clock	CLK
Input	I
Output	O
Interrupt	INT
Data	DATA
Size	SIZE
Divider	DIV
Prescaler	PSC
Multiplier	MUL
Period	PRD

Table 3 Full Names and Abbreviations of Modules

Full Name	Abbreviation
Reset and Clock Management	RCM
Clock Recovery System	CRS
Power Management Unit	PMU
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
Universal Synchronous/ Asynchronous Receiver Transmitter	USART
Serial Peripheral Interface	SPI
Controller Area Network with Flexible Data Rate	CAN FD
Analog-to-Digital Converter	ADC
Cyclic Redundancy Check Calculation Unit	CRC
Error Report	ERP
Real-Time Clock	RTC
Temperature Sensor	T-Sensor

Full Name	Abbreviation
Random Number Generator	TRNG
Advanced Encryption Standard	AES256
Secure Hash Algorithm 256-bit Version	SHA256

2 System Architecture

2.1 Full Name and Abbreviation Description of Terms

Table 4 Full Names and Abbreviations of Terms

Full Name	Abbreviation
Advanced High-Performance Bus	AHB
Advanced Peripheral Bus	APB

2.2 System Architecture Block Diagram

The main system mainly consists of two master modules and four slave modules. The main modules are Arm® Cortex®-M0+ core and general-purpose DMA. The slave modules are internal SRAM, internal flash memory Flash, AHB2 bus connecting all GPIO ports, and AHB1/APB bridges on AHB1 bus, among which, AHB1/APB bridges connect all peripherals.

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

Figure 1 G32A1085 A1065 A1045 System Architecture Block Diagram

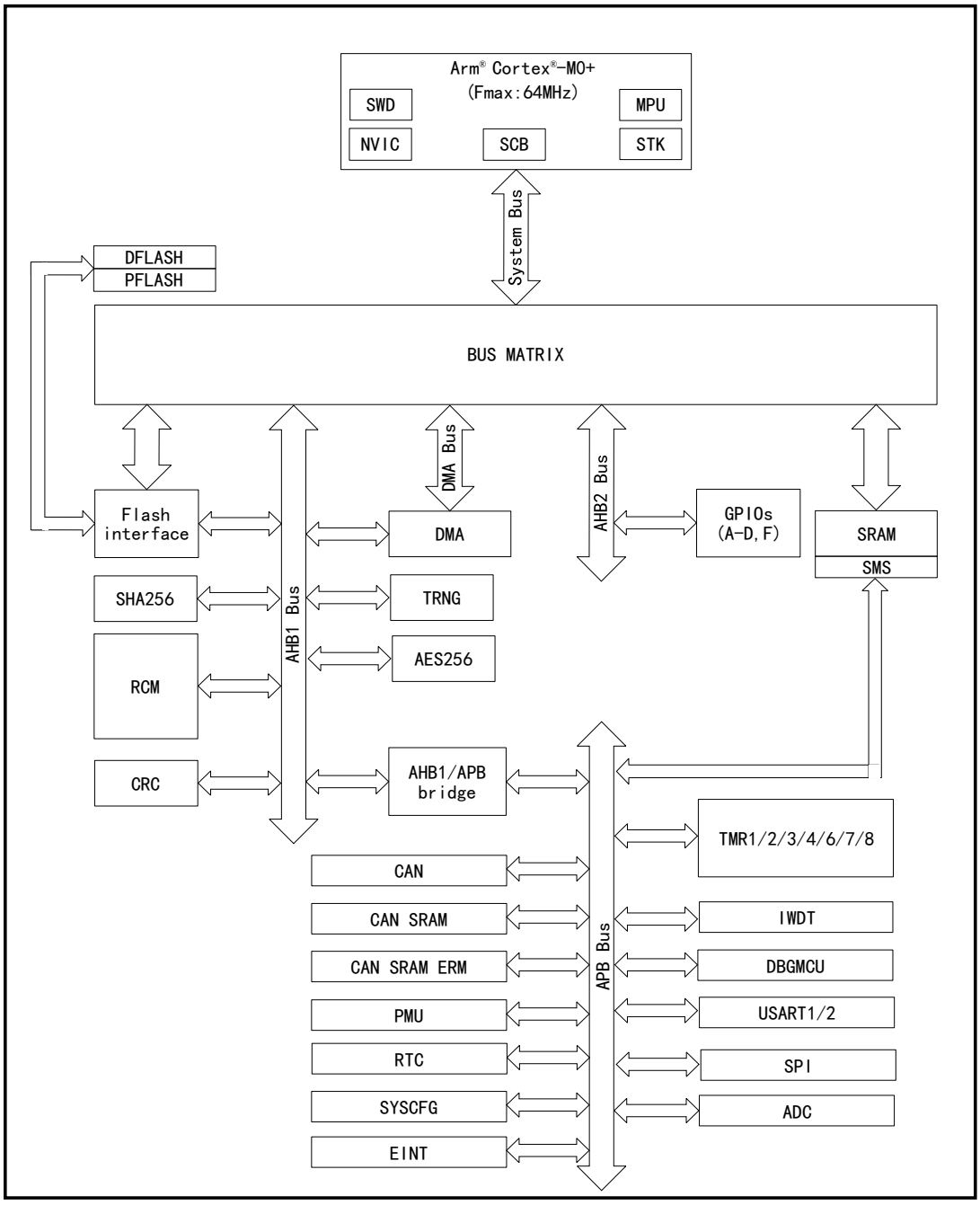


Table 5 Bus Name

Name	Description
System bus	Connect the system bus (peripheral bus) of Arm® Cortex®-M0+ core and the bus matrix.
DMA bus	Connect the AHB master control interface of DMA and the bus matrix.
Bus matrix	Coordinate the access of the core and DMA; consist of CPU AHB, system bus, DMA bus and FMC, SRAM, AHB2 and AHB1/APB bridges. AHB peripheral is connected with the system bus through the bus matrix and is allowed to access DMA.
AHB/APB bridge	The bridge provides synchronous connection between AHB and APB buses. 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 4 GB. 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, and bus peripherals (including AHB and APB peripherals). For specific information of various addresses, refer to the data manual of the corresponding model.

2.3.1 Embedded SRAM

Built-in static SRAM. It allows access by byte, half word (16 bits) or full word (32 bits). The start address of SRAM is 0x2000 0000 (Not include CAN-specific SRAM; CAN-specific SRAM only supports full-word access).

3 Flash Memory

This manual is only applicable to G32A1085 A1065 A1045 series products, mainly introducing the Flash storage structure, read, erase, write, read/write protection, unlock/lock characteristics, and the function description of related registers.

3.1 Full Name and Abbreviation Description of Terms

Table 6 Full Names and Abbreviations of Terms

Full Name	Abbreviation
Flash Memory Controller	FMC
Error Correction Code	ECC

3.2 Flash Memory Characteristics

(1) Flash memory structure:

- It is divided into PFlash memory area, DFlash memory area, and information block.
- PFlash memory capacity is up to 256 KB.
- DFlash memory capacity is up to 32 KB.
- Page size is 512 B.
- The information block is divided into system memory and option byte area.
- The system storage area stores the 96-bit unique UID and Flash capacity information.
- The capacity of the option byte area is 24 bytes.

(2) Functional description:

- Read Flash
- Page/Mass erase Flash
- Write Flash
- Read/Write protection Flash
- Both the PFlash and DFlash memory areas are equipped with ECC function, which can correct 1-bit errors and detect 2-bit errors.
- Configure option bytes

3.3 Flash Memory Structure

Table 7 Flash Memory Structure for G32A1085 Series Product

Block	Name	Address area	Size (bytes)	Sector
PFlash	Page 0	0x0800 0000 – 0x0800 01FF	512	Sector 0

Block	Name	Address area	Size (bytes)	Sector
PFlash	Page 1	0x0800 0200 – 0x0800 03FF	512	
PFlash	Page 2	0x0800 0400 – 0x0800 05FF	512	
PFlash	Page 3	0x0800 0600 – 0x0800 07FF	512	
PFlash	Page 4	0x0800 0800 – 0x0800 09FF	512	
PFlash	Page 5	0x0800 0A00-0x0800 0BFF	512	
PFlash	…	…	…	
PFlash	Page 15	0x0800 1E00 – 0x0800 1FFF	512	
PFlash	…	…	…	…
PFlash	Page 496	0x0803 E000 – 0x0803 E1FF	512	
PFlash	…	…	…	
PFlash	Page 506	0x0803 F400 – 0x0803 F5FF	512	
PFlash	Page 507	0x0803 F600 – 0x0803 F7FF	512	Sector 31
PFlash	Page 508	0x0803 F800 – 0x0803 F9FF	512	
PFlash	Page 509	0x0803 FA00 – 0x0803 FBFF	512	
PFlash	Page 510	0x0803 FC00 – 0x0803 FDFF	512	
PFlash	Page 511	0x0803 FE00 – 0x0803 FFFF	512	
DFlash	Page 0	0x0804 0000 – 0x0804 01FF	512	Sector 0
DFlash	Page 1	0x0804 0200 – 0x0804 03FF	512	
DFlash	…	…	…	…
DFlash	Page 62	0x0804 7C00 – 0x0804 7DFF	512	Sector 31
DFlash	Page 63	0x0804 7E00 – 0x0804 7FFF	512	
Information block	System memory area	0x1FFF E800 – 0x1FFF F5FF	3.5K	-
Information block	Option byte	0x1FFF F800 – 0x1FFF F817	24	-

Table 8 Flash Memory Structure for G32A1065 Series Product

Block	Name	Address area	Size (bytes)	Sector
PFlash	Page 0	0x0800 0000 – 0x0800 01FF	512	
PFlash	Page 1	0x0800 0200 – 0x0800 03FF	512	
PFlash	Page 2	0x0800 0400 – 0x0800 05FF	512	
PFlash	Page 3	0x0800 0600 – 0x0800 07FF	512	Sector 0
PFlash	Page 4	0x0800 0800 – 0x0800 09FF	512	
PFlash	Page 5	0x0800 0A00-0x0800 0BFF	512	

Block	Name	Address area	Size (bytes)	Sector
PFlash	…	…	…	
PFlash	Page 15	0x0800 1E00 – 0x0800 1FFF	512	
PFlash	…	…	…	…
PFlash	Page 255	0x0801 FE00 – 0x0801 FFFF	512	Sector 15
DFlash	Page 0	0x0804 0000 – 0x0804 01FF	512	Sector 0
DFlash	Page 1	0x0804 0200 – 0x0804 03FF	512	
DFlash	…	…	…	…
DFlash	Page 62	0x0804 7C00 – 0x0804 7DFF	512	Sector 31
DFlash	Page 63	0x0804 7E00 – 0x0804 7FFF	512	
Information block	System memory area	0x1FFF E800 – 0x1FFF F5FF	3.5K	-
Information block	Option byte	0x1FFF F800 – 0x1FFF F817	24	-

Table 9 Flash Memory Structure for G32A1045 Series Product

Block	Name	Address area	Size (bytes)	Sector
PFlash	Page 0	0x0800 0000 – 0x0800 01FF	512	Sector 0
PFlash	Page 1	0x0800 0200 – 0x0800 03FF	512	
PFlash	Page 2	0x0800 0400 – 0x0800 05FF	512	
PFlash	Page 3	0x0800 0600 – 0x0800 07FF	512	
PFlash	Page 4	0x0800 0800 – 0x0800 09FF	512	
PFlash	Page 5	0x0800 0A00 – 0x0800 0BFF	512	
PFlash	…	…	…	
PFlash	Page 15	0x0800 1E00 – 0x0800 1FFF	512	
PFlash	…	…	…	…
PFlash	Page 127	0x0800 FE00 – 0x0800 FFFF	512	Sector 7
DFlash	Page 0	0x0804 0000 – 0x0804 01FF	512	Sector 0
DFlash	Page 1	0x0804 0200 – 0x0804 03FF	512	
DFlash	…	…	…	…
DFlash	Page 14	0x0804 1C00 – 0x0804 1DFF	512	Sector 15
DFlash	Page 15	0x0804 1E00 – 0x0804 1FFF	512	
Information block	System memory area	0x1FFF E800 – 0x1FFF F5FF	3.5K	-

Block	Name	Address area	Size (bytes)	Sector
Information block	Option byte	0x1FFF F800 – 0x1FFF F817	24	-

3.4 Function Description of Flash Memory

3.4.1 Read Flash

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

Wait Cycle and Read Modes

Different read modes require different wait cycles according to the system clock frequency.

Three different read modes are provided:

(1) Normal Read: It is the default read mode.
- 0 wait cycle: 0<system clock≤24 MHz
- 1 wait cycle: 24 MHz<system clock≤48 MHz
- 2 wait cycles: 48 MHz<system clock≤64 MHz

(2) VREAD0 Read: A special read mode, is slower and more stringent than the normal read mode, employed during read verification after Flash programming.
- 5 wait cycles: 0<system clock≤24 MHz
- 10 wait cycles: 24 MHz<system clock≤48 MHz
- 15 wait cycles: 48 MHz<system clock≤64 MHz

(3) VREAD1 Read: A special read mode, is slower and more stringent than the normal read mode, employed during read verification after Flash erasure.
- 5 wait cycles: 0<system clock≤24 MHz
- 10 wait cycles: 24 MHz<system clock≤48 MHz
- 15 wait cycles: 48 MHz<system clock≤64 MHz

Procedure for entering special read mode (Using VREAD0 as an example):

(1) Set the FMC_CTRL1[VREAD0_EN] bit;

(2) Wait for the BUSYF signal to be low (OCF is set); confirm that the FMC_CTRL1[VREAD0_STS] status bit is set. Subsequently, all read operations initiated to the Flash will be performed in VREAD0 mode.

Procedure for exiting special read mode (Using VREAD0 as an example):

(1) Clear the FMC_CTRL1[VREAD0_EN] bit;

(2) Wait for the BUSYF signal to be low (OCF is set); confirm that the FMC_CTRL1[VREAD0_STS] status bit is cleared. The Flash has now exited VREAD0 read mode.

Note:

(1) Only one read mode can be enabled at the same time. The default mode is normal read mode.

(2) When switching between the two VREAD modes (for example, from VREAD0 to VREAD1), you must first exit VREAD0 and then enable VREAD1. You must not set VREAD1_EN and clear VREAD0_EN simultaneously.

Prefetch buffer

The prefetch buffer improves read speed. It is disabled by default upon each reset.

ECC Check

An ECC algorithm that corrects single-bit errors and detects double-bit errors can be used to verify data correctness, so as to enhancing data reliability. This feature is enabled by default after a reset.

Once this feature is enabled, if an uncorrectable ECC error is detected, the error flag DBFIFLG is set. If DBFIEN is enabled, an interrupt can be generated.

After an uncorrectable ECC error occurs, the address of the last error is recorded in the FLASH_ECC_ADDR register. When DBFIFLG is cleared, the error address is also cleared simultaneously.

3.4.2 Main Memory Block

3.4.2.1 Erase Main Memory Block

FMC supports page erase and mass erase (erase all) to initialize the contents of Flash to high level. Before writing to Flash, users are advised to erase the write address page. If the write address does not contain all 1, a programming error will be triggered.

Page Erase

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

After the correct page erase (or 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 should note that the page selected for erase must be a valid page (the address of PFlash or DFlash that is valid and not write-protected).

Page Erase Procedure:

(1) Unlock the FLASH and check the BUSYF flag to ensure no ongoing FLASH operation.

(2) Set the PAGEERA bit.

(3) Write any address within the target erase page to the FMC_ADDR register.

(4) Set the STA bit.

(5) Wait for the BUSYF signal to be low, then perform a read verification on the erased address values.

Mass Erase

A mass erase operation can erase the entire contents of the PFlash, the DFlash, or both. Users must pay attention when using this function to avoid the loss of critical data.

Page Erase Procedure:

(1) Unlock the FLASH and check the BUSYF flag to ensure no ongoing FLASH operation.

(2) Set the MASSERA bit.

(3) Configure the MER_TYPE register to select the mass erase areas.

(4) Set the STA bit.

(5) Wait for the BUSYF signal to be low, then perform a read verification on the erased address values.

3.4.2.2 Write Main Memory Block

The write address for FMC operations to the FLASH memory area must be 64-bit aligned; otherwise, the PAEF will be set. The programming length can be configured via PROGLEN, with a minimum of 64 bits (8 bytes) of data.

Note: If consecutively writing multiple double words, the operation must not cross the 1/2 boundary of a page. For example, when performing consecutive write operations to Page 0 of PFLASH, the write address must not cross the address 0x0800 0100.

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 will be invalid and a write protection error will be triggered (WPEF bit of FMC_STS is set to "1").

Programming Procedure:

(1) Unlock the FLASH and check the BUSYF flag to ensure no ongoing FLASH operation.

(2) Set the PG bit.

(3) Write the programming start address to the FMC_ADDR register (ensure it is double-word aligned).

(4) Configure the programming length PROGLEN, write the data to be programmed sequentially into the FLASH_PROG_DATAx registers (for example, to program the data 0x12345678_22345678, first write 0x22345678 to FLASH_PROG_DATA0, then write 0x12345678 to FLASH_PROG_DATA1. If more data needs to be written, continue writing to the subsequent FLASH_PROG_DATA registers in order).

(5) Set the STA bit.

(6) Wait for the BUSYF signal to be low, then perform a read verification on the programmed address values.

3.4.2.3 Read/Write Protection of Main Memory Block

Read/Write protection of the Flash is used to prevent invalid reading/modification of the memory area code or data, and it is controlled by the read/write protection configuration byte of option byte. For the G32A1085 A1065 A1045 series products, the write protection base unit for PFLASH is 16 pages (8 KB), and for DFLASH, it is 2 pages (1 KB).

Read Protection

The read protection has three levels, namely, Level 0, Level 1 and Level 2, which are specifically described as follows:

Table 10 Difference among Read Protection Levels

Category	READPROT	Description
Level 0	0xAA	The main memory block and option byte are erasable, writable and readable.
Level 1	Other values except 0xAA and 0xCC	User Mode: Allows erase, write, and read operations to the main memory block and option bytes. Debug & SRAM Execution Mode: Access to the main memory block is prohibited (except for mass erase operations). Option bytes can be erased, written, and read. When the protection level is modified to Level 0, a mass erase of the main memory block is performed first.
Level 2	0xCC	Mode where Debug & SRAM Execution are disabled: Erase, write, and read operations are allowed for the main memory block. Reading the Option Bytes is allowed. Writing to Option Bytes (except for the Read Protection Level field) is allowed. Erasing the Option Bytes is prohibited.

Write protection

Write protection for specific pages in the main memory block can be configured by setting the respective values in the write protection option bytes (PWRP and DWRP). Once write protection is enabled, the content of the corresponding pages in the main memory block cannot be modified by any means.

3.4.2.4 Unlock/Lock Main Memory Block

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

- KEY1=0x4567 0123
- KEY2=0xCDEF 89AB

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. Users can also lock FMC by software by writing "1" to LOCK bit of the control register 2 (FMC_CTRL2).

In each Flash programming operation, 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.

Page Erase Procedure:

(1) Check the BUSYF flag to ensure no ongoing FLASH operation.

(2) Unlock the option bytes area and wait for OBWEN to be set.

(3) Set the OBE bit.

(4) Set the STA bit.

(5) Wait for the BUSYF signal to be low, then perform a read verification on the erased address values.

3.4.3.2 Write Option Byte

All 12 configurable bytes of the option bytes support write function.

Programming Procedure:

(1) Check the BUSYF flag to ensure no ongoing FLASH operation.

(2) Unlock the option bytes area and wait for OBWEN to be set.

(3) Set the OBP bit.

(4) Write the programming start address to the FMC_ADDR register (ensure it is double-word aligned).

(5) Configure the programming length PROGLEN and sequentially write the data to be programmed into the FLASH_PROG_DATAx registers (the hardware ensures complementary data writing).

(6) Set the STA bit.

(7) Wait for the BUSYF signal to be low, then perform a read verification on the programmed address values.

3.4.3.3 Write-protect option byte

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 the KEY value is written to FMC_OBKEY register rather than FMC_KEY register. Users must pay special attention that after modifying the values of the option bytes, a forced reload (by setting the OBLOAD bit) or a power-on reset is required for the new settings to take effect.

3.4.4 Functional Description of Option Byte

The option byte provides some optional functions for users, and it mainly consists of 12 configurable bytes and corresponding complementary codes. After the system is powered on, the option byte area will be reloaded to the FMC_ OBCS and FMC_WRTPROT0/1 registers (the option bytes will only take effect each time they are reloaded to FMC register). In the process of reloading, if a certain configurable byte does not match its inverse code, an option byte error (OBE bit of FMC_register is set to "1") will be triggered, and this byte will be set to "0xFF". The 24-byte information of the option byte area is shown in the table below.

Table 11 Option Bytes

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F800	READPROT	0xAA	R/W	Read protection configuration Bit [7:0]: READPROT 0xAA: Level 0 0xCC: Level 2 Others: Level 1
0x1FFF F801	nREADPROT	0x55	R	READPROT complementary code
0x1FFF F802	UOB	0xFF	R/W	User option byte Bit 0: WDTSEL 0: Hardware enabling watchdog 1: Software enabling watchdog (default) Bit 1: nRSTSTOP 0: Reset occurs when entering the Stop mode 1: Reset does not occur when entering the Stop mode Bit 2: nRSTSTANDBY 0: Reset occurs when entering the Standby mode 1: Reset does not occur when entering the Standby mode Bit 3: nBOOT0 Bit 4: nBOOT1 Bit 5: VDDA_MONITOR 0: V_{DDA} power supply detector is disabled 1: V_{DDA} power supply detector is enabled Bit 6: Reserved Bit 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	Data1 complementary code
0x1FFF F808	WRP0	0xFF	R/W	PFLASH write protection set to 0
0x1FFF F809	nWRP0	0x00	R	WRP0 complementary code
0x1FFF F80A	WRP1	0xFF	R/W	PFLASH write protection set to 1
0x1FFF F80B	nWRP1	0x00	R	WRP1 complementary code
0x1FFF F80C	WRP2	0xFF	R/W	PFLASH write protection set to 2
0x1FFF F80D	nWRP2	0x00	R	WRP2 complementary code
0x1FFF F80E	WRP3	0xFF	R/W	PFLASH write protection set to 3

Address	Option byte	Initial value	R/W	Functional description
0x1FFF F80F	nWRP3	0x00	R	WRP3 complementary code
0x1FFF F810	WRP4	0xFF	R/W	DFLASH write protection set to 0
0x1FFF F811	nWRP4	0x00	R	WRP4 complementary code
0x1FFF F812	WRP5	0xFF	R/W	DFLASH write protection set to 1
0x1FFF F813	nWRP5	0x00	R	WRP5 complementary code
0x1FFF F814	WRP6	0xFF	R/W	DFLASH write protection set to 2
0x1FFF F815	nWRP6	0x00	R	WRP6 complementary code
0x1FFF F816	WRP7	0xFF	R/W	DFLASH write protection set to 3
0x1FFF F817	nWRP7	0x00	R	WRP7 complementary code

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

Table 12 Write Protection (WRPx) Description of Main Memory Block

Product Capacity	Functional description
G32A1085 A1065 A1045	Each bit in the WRPx register controls the write protection for an 8 KB (16-page) or 1 KB (2-page) address range in the main memory block. 0: Enable write protection 1: Dissable write protection WRP0: Page 0-127 WRP1: Page 128-255 (only for G32A1085 and G32A1065) WRP2: Page 256-383 (only for G32A1085) WRP3: Page 384-511 (only for G32A1085) WRP4: Page 0-15 WRP5: Page 16-31 WRP6: Page 32-47 (only for G32A1085 and G32A1065) WRP7: Page 48-63 (only for G32A1085 and G32A1065)

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

3.5 Register Address Mapping

Base address: 0x4002 2000

Table 13 FMC Register Address Mapping

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

Register name	Description	Offset address
FMC_STS	Status register	0x0C
FMC_CTRL2	Control register 2	0x10
FMC_ADDR	Address register	0x14
FMC_OBCS	Option byte control/status register	0x1C
FMC_WRTPROT0	Write protection register 0	0x20
FMC_WRTPROT1	Write protection register 1	0x24
FMC_ECC_ADDR	ECC error address register	0x30
FMC_PROG_DATAy	Programming data register	0x40+4y

3.6 Register Functional Description

3.6.1 Control Register 1 (FMC_CTRL1)

Offset address: 0x00

Reset value: 0x0001 0000

Field	Name	R/W	Description
3:0	WS	R/W	Wait State Configure 000: 0 wait cycle, 0<system clock≤24 MHz 001: 1 wait cycle, 24 MHz<system clock≤48 MHz 010: 2 wait cycles, 48 MHz<system clock≤64 MHz Others: Reserved
4	PBEN	R/W	Prefetch Buffer Enable 0: Disable 1: Enable
5	PBSF	R	Prefetch Buffer Status Flag 0: Closed 1: Open
7:6	Reserved		
8	VREAD0_EN	R/W	VREAD0 read mode enabled (VREAD0 Enable) 0: Disable 1: Enable
9	VREAD0_STS	R	VREAD0 read mode enabled status (VREAD0 Status) When VREAD0 read mode is enabled, this bit will be set by hardware.
11:10	Reserved		
12	VREAD1_EN	R/W	VREAD1 read mode enabled (VREAD1 Enable) 0: Disable 1: Enable
13	VREAD1_STS	R	VREAD1 read mode enabled status (VREAD1 Status) When VREAD1 read mode is enabled, this bit will be set by hardware.
15:14	Reserved		

Field	Name	R/W	Description
16	ECCEN	R/W	ECC Enabling (ECC Enable) 0: Disable 1: Enable
31:17	Reserved		

3.6.2 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.

3.6.3 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.

3.6.4 Status 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 memory operation is in progress. It is set by hardware when the flash operation starts and is cleared when the operation completes or if an error occurs.
1	Reserved		
2	PEF	RC_W1	Programming Error Flag This bit will be set by hardware when the address value is not "0xFFFF" before being edited. Write "1" to this bit to clear it. The STA bit in the FMC_CTRL2 register should be cleared before initiating a new programming operation.
3	PAEF	RC_W1	Programming Alignment Error Flag This flag is set by hardware when the programming address is not double-word aligned. It is cleared by writing "1" to it.
4	WPEF	RC_W1	Write Protection Error Flag This bit will be set by hardware when programming the write protection address in FLASH.

Field	Name	R/W	Description
5	OCF	RC_W1	Operation Complete Flag This bit will be set by hardware when read/write operation in FLASH is completed.
15:6	Reserved		
16	DBFIFLG	R/W	Double Bit Fault Interrupt Flag This bit indicates that an uncorrectable ECC error is detected when reading FLASH. Writing 1 can clear this bit, and writing 0 is invalid. 0: No double-bit fault is detected during valid storage read access of FLASH. 1: A double-bit fault is detected during valid storage read access of FLASH.
31:17	Reserved		

3.6.5 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 BUSYF bit of FMC_STS.
7	LOCK	R/W	Lock This bit can be only written to 1, and when this bit is set to 1, it means that FMC and CTRL2 registers are locked. This bit is cleared by hardware upon detection of the unlock sequence.
8	Reserved		
9	OBWEN	R/W	Enable Option Byte Write When this bit is set to 1, the option byte can be programmed.
10	ERRIE	R/W	Error interrupt Enable 0: Disable interrupt 1: Enable interrupt When PEF=1 or WPEF=1 of FMC_STS, setting this bit will generate an interrupt.

Field	Name	R/W	Description
11	Reserved		
12	OCIE	R/W	Operation Complete Interrupt Enable 0: Disable operation completion interrupt 1: Enable operation completion interrupt If this bit is set, an interrupt can be generated when STS_OCF = 1.
13	OBLOAD	R/W	Force Option Byte Load When this bit is set to 1, force to reload the option byte to generate system reset. 0: Idle 1: Force to load
15:14	Reserved		
16	DBFIEN	R/W	Double Bit Fault Interrupt Enable When an uncorrectable ECC fault is detected during FLASH valid storage read access, DBFIEN control will generate an interrupt. 0: Disable 1: Enable. As long as the DBFIFLG bit is set, an interrupt request will be generated.
17	OBRIE	R/W	Option Byte Register Store Error Interrupt Enable 0: Disable 1: Enabled. As long as the OPTRERR flag is set, an interrupt request is generated.
19:18	MER_TYPE	R/W	Mass Erase Area Selection 00: Select PFLASH 01: Select DFLASH 1x: Select PFLASH and DFLASH
21:20	PROGLEN	R/W	Programming Length Configuration One time writing of 64 bits*(2^ PROGLEN)
23:22	Reserved		
31:24	SCR	W	FLASH Initialization Control When it is configured to 8'h66, the FLASH initialization is re-executed. If it is configured to any other value, there are no effects.

3.6.6 Address Register (FMC_ADDR)

Offset address: 0x14

Reset value: 0x0000 0000

This register must be configured by software before initiating any FLASH erase or write operation.

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 erase, this bit is written to the page to be erased.

3.6.7 Option Byte Control/Status Register (FMC_OBCS)

Offset address: 0x1C

Reset value: 0xXXXX XX0X

The reset value of the register is related to the value written in the 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 inverse code.

Field	Name	R/W	Description
0	OBE	R	Option Byte Error 0: No corresponding 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
2:1	READPROT	R	Read Protection Display which level of read protection is enabled. The level is 1 when Bit 1 is set, and 2 when Bit 2 is set. 00: Level 0 01: Level 1 1x: Level 2
3	SCRKERR	R	FLASH Initialization Key Error 0: No corresponding error 1: Indicates the initialization key stored in the FLASH is incorrect.
4	SCRECCERR	R	Uncorrectable ECC Error during FLASH Initialization 0: No corresponding error 1: An uncorrectable ECC error is detected during FLASH initialization.
5	OPTRERR	R	Option Byte Store Error 0: No corresponding error 1: A non-complementary error is detected in the option bytes after they were loaded.
6	OPTECCERR	R	Uncorrectable ECC Error in Option Bytes 0: No corresponding error 1: An uncorrectable ECC error is detected in the option bytes after they are loaded.
7	Reserved		
8	WDTSEL	R	Select Watchdog 0: Hardware watchdog 1: Software watchdog
9	RSTSTOP	R	nReset in STOP Mode 0: Generate 1: Not generate
10	RSTSTDB	R	nReset in STANDBY Mode 0: Generate 1: Not generate
11	nBOOT0	R	Configure nBOOT0 Mode
12	nBOOT1	R	Configure nBOOT1 Mode

Field	Name	R/W	Description
15:13	Reserved		
23:16	DATA0	R	Data 0
31:24	DATA1	R	Data 1

3.6.8 Write Protection Register 0 (FMC_WRTPROT0)

Offset address: 0x20

Reset value: 0xXXXX XXXX (the reset value depends on the programming value in option byte)

Field	Name	R/W	Description
31:0	PWRTPROT0	R	PFLASH Write Protection Each bit provides the write protection of 16 pages. For details, refer to the corresponding table-Write Protection (WRPx) Description of Main Memory Block. 0: Valid 1: Invalid

3.6.9 Write Protection Register 1 (FMC_WRTPROT1)

Offset address: 0x24

Reset value: 0xXXXX XXXX (the reset value depends on the programming value in option byte)

Field	Name	R/W	Description
31:0	DWRTPROT1	R	DFLASH Write Protection Each bit provides the write protection of 2 pages. For details, refer to the corresponding table-Write Protection (WRPx) Description of Main Memory Block. 0: Valid 1: Invalid

3.6.10 ECC Address Register (FMC_ECC_ADDR)

Offset address: 0x30

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	FMC_ECC_ADDR	R	Address of the Last Detected Uncorrectable ECC Error When the ECC error flag is cleared, this register is also cleared

3.6.11 Programming Data Register (FMC_PROG_DATAy) (y=0…15)

Offset address: 0x40+4y

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	FMC_PROG_DATAy	W	Data Written to Flash Values Written to FLASH during PFLASH or DFLASH Programming

4 Static Memory Subsystem (SMS)

4.1 Full Name and Abbreviation Description of Terms

Table 14 Full Names and Abbreviations of Terms

Full Name	Abbreviation
Static Memory Subsystem	SMS
Software	SW
Hardware	HW
Interface	I/F or IF
AHB Slave Interface	AHBS IF
AHB interface	AHB IF
Hardware-oriented, Pre-synthesis User-defined	CFG HW
Software-oriented, post-synthesis user-programmed through SW I/F	PRG SW
Arbiter	ARB
Decoder	DEC
Store Queue	STQ
Most Significant Bit	MSB
Built-in self test	BIST
Error Correction Code	ECC
Normal Read	NR
Read-Modify-Write	RMW

4.2 Functional Description

4.2.1 ECC

The SMS (SRAM Memory System) provides ECC (Error Correction Code) protection for the SRAM, encompassing error detection, single-bit error correction, and double-bit error detection. Each address has additional check bits appended. The 32-bit data is configured with 7 check bits.

When the ECC feature is enabled, the SMS internally calculates the check bits during every write operation. If the number of valid data bytes in a write operation does not match the memory's bit width, the SMS will first block the write and then perform a Read-Modify-Write (RMW) operation. This process introduces an additional cycle of latency.

Note: When ECC is enabled and the write transfer size is smaller than the memory data bit width, an RMW transfer is initiated to generate the correct ECC check bits.

Example: When the AHB slave interface writes a single byte of data to a 32-bit memory interface, the 32-bit memory interface will return one valid byte and three invalid bytes. In this case, the system executes an RMW operation. The SMS merges the read data with the write data and generates the ECC check bits for the complete word.

For every read operation (including RMW operations), any ECC errors are reported and logged. The reported ECC errors can generate a maskable system interrupt, and the valid address and ECC error status are recorded. If a single-bit error is detected, error correction can be programmed for both NR and RMW.

The SMS can inject error bits during SRAM accesses for diagnostic purposes. The INSERT_ERR register can be configured to inject single-bit, double-bit, or triple-bit errors.

If the number of error bits exceeds two (exceeding the correction and detection capability of the ECC logic), the data will not be corrected.

4.2.2 Programming Guide

The SMS configuration registers (including SMS_INTRMn and SMS_ECC_CFG) can be configured at any time without specific sequence restrictions. However, since each configuration register operates in the PCLK domain, whose frequency may be significantly lower than that of the HCLK domain signals, it is recommended to adopt a write-then-read strategy.

For certain configuration registers related to timing or architecture, such as SMS_ECC_CFG, they should not be configured if there are pending AHB transactions. It is advised to insert a barrier instruction before and after the register configuration instruction.

When ECC is enabled, the SMS reports interrupts without blocking AHB transactions. Therefore, new interrupts may be generated during or after the interrupt handling process.

The configuration registers SMS_STAT, SMS_ECC_LOGn, SMS_INTRRn, and SMS_INTRSn operate in the HCLK domain. Due to their high-speed operation, rapid register overwriting may occur.

4.3 Register Address Mapping

Table 15 SMS Register Address Mapping

Register name	Description	Offset address
SMS_ECC_CFG	Error correction configuration register	0x004
SMS_STAT	Status register	0x008
SMS_ECC_LOGn	Error address register	0x100

Register name	Description	Offset address
SMS_INTRRn	Interrupt flag clear register	0x140
SMS_INTRMn	Interrupt mask register	0x180
SMS_INTRSn	Interrupt flag register	0x1C0

4.4 Register Functional Description

4.4.1 Error Correction Configuration Register (SMS_ECC_CFG)

Offset address: 0x004

Reset value: 0x0000 0001

This register configures the interrupt notification function.

Field	Name	R/W	Description
0	ECC_EN	R/W	Enable ECC Detection 0: Disable 1: Enable
1	RD_COR_EN	R/W	ECC Correction Enable 0: Disable 1: Enable
2	RMW_COR_EN	R/W	Enable hardware read ECC error detection during ECC half-word and byte access 0: Disable 1: Enable
4:3	INSERT_ERR	R/W	Error Insertion at ECC Encoder Output (Insert Error) 00: No error inserted. 01: A single-bit error is inserted on RAM data [0]. 10: A double-bit error is inserted on RAM data [1:0]. 11: A triple-bit error is inserted on RAM data [2:0]. This is only valid when SMS_ECC_CFG[ECC_EN] is set to 1.
31:5	Reserved		

4.4.2 State Register (SMS_STAT)

Offset address: 0x008

Reset value: 0x0000 0000

If any bit in this register becomes active, an interrupt signal is generated.

Reading this register clears the interrupt signal.

Field	Name	R/W	Description
0	RD_ERR_DET	R	Read Error Detection This bit is set when an error is detected in the read data (RD) and the error is not masked.
1	RD_ERR_SIN	R	Read Single Bit Error Detection This bit is set when a single-bit error is detected in the read data (RD) and the error is not masked.

Field	Name	R/W	Description
2	RD_ERR_DBL	R	Read Double Bits Error Detection This bit is set when a double-bit error is detected in the read data (RD) and the error is not masked.
3	RMW_ERR_DET	R	RMW ECC Error Detection This bit is set when an ECC error is detected during an RMW (Read-Modify-Write) operation and the error is not masked.
4	RMW_ERR_SIN	R	RMW Read Single Bit Error Detection This bit is set when a single-bit error is detected during an RMW operation and the error is not masked.
5	RMW_ERR_DBL	R	RMW Double Bits Error Detection This bit is set when a double-bit error is detected during an RMW operation and the error is not masked.
15:6	Reserved		
16	IF_INTR	R	Interface Interrupt Each bit in this field indicates whether its corresponding interface has generated an interrupt.
31:17	Reserved		

4.4.3 Error Address Register (SMS_ECC_LOGn)

Offset address: 0x100

Reset value: 0x0000 0000

The ERP Memory Error Address Register is used to capture the address of the last ECC event in the memory.

This register is read-only and any attempt to write to it will be ignored.

After an ECC event occurs, the reserved bits in the SMS_ECC_LOGn registers may not be zero; please ignore them. When reading the address, use bit-operation to eliminate their effects.

Field	Name	R/W	Description
6:0	Reserved		
19:7	ADDR	R	Error Address Memory Error Address: This field records the error system address (at 32-bit address granularity) of the last ECC event in the memory. Error Address= 0x2000 0000 + ADDR*4
31:20	Reserved		

4.4.4 SMS Interrupt Flag Reset Register (SMS_INTRRn)

Offset address: 0x140

Reset value: 0x0000 0000

The SMS_INTRRn register can be reset by a system reset or by a write operation. Resetting this register will clear all bits in SMS_STAT and SMS_INTRSn registers.

Field	Name	R/W	Description
0	RD_ERR_DET	R/W	Read ECC Error Detection This bit is set when an Error Correction Code (ECC) error is detected in a read operation on this interface.
1	RD_ERR_SIN	R/W	Read Single Bit Error Detection This bit is set when a single-bit error occurs in the Read Data (RD) on this interface.
2	RD_ERR_DBL	R/W	Read Double Bits Error Detection This bit is set when a double-bit error occurs in the Read Data (RD) on this interface.
3	RMW_ERR_DET	R/W	RMW Error Detection This bit is set when an error is detected in a Read-Modify-Write (RMW) operation on this interface.
4	RMW_ERR_SIN	R/W	RMW Read Single Bit Error Detection This bit is set when a single-bit error is detected during an RMW operation on this interface.
5	RMW_ERR_DBL	R/W	RMW Double Bits Error Detection This bit is set when a double-bit error is detected during an RMW operation on this interface.
31:6	Reserved		

4.4.5 Interrupt Mask Register (SMS_INTRMn)

Offset address: 0x180

Reset value: 0x0000 003F

Field	Name	R/W	Description
0	RD_ERR_DET	R/W	Read Error Detection Interrupt This option masks interrupt generation when an error is detected during a read operation on this interface.
1	RD_ERR_SIN	R/W	Read Single Bit Error Detection Interrupt This option masks interrupt generation when a single-bit error is detected during a read operation on this interface.
2	RD_ERR_DBL	R/W	Read Double Bits Error Detection Interrupt This option masks interrupt generation when an ECC double-bit error is detected during a read operation on this interface.
3	RMW_ERR_DET	R/W	RWM Error Detection This option masks interrupt generation when an error is detected during a Read-Modify-Write (RMW) operation on this interface.
4	RMW_ERR_SIN	R/W	RMW Read Single Bit Error Detection This option masks interrupt generation when a single-bit error is detected during a Read-Modify-Write (RMW) operation on this interface.
5	RMW_ERR_DBL	R/W	RMW Double Bits Error Detection This option masks interrupt generation when a double-bit error is detected during a Read-Modify-Write (RMW) operation on this interface.
31:6	Reserved		

4.4.6 Interrupt Flag Register (SMS_INTRSn)

Offset address: 0x1C0

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	RD_ERR_DET	R	Read ECC Error Detection This bit is set when an Error Correction Code (ECC) error is detected during the read operation on this interface and the error is not masked.
1	RD_ERR_SIN	R	Read Single Bit Error Detection This bit is set when a single-bit error occurs during the read operation on this interface and the error is not masked.
2	RD_ERR_DBL	R	Read Double Bits Error Detection This bit is set when a double-bit error occurs during the read operation on this interface and the error is not masked.
3	RMW_ERR_DET	R	RWM Error Detection This bit is set when an error correction code error is detected during the Read-Modify-Write (RMW) operation on this interface and the error is not masked.
4	RMW_ERR_SIN	R	RMW Read Single Bit Error Detection This bit is set when a single-bit error is detected during an RMW operation on this interface and the error is not masked.
5	RMW_ERR_DBL	R	RMW Double Bits Error Detection This bit is set when a double-bit error is detected during an RMW operation on this interface and the error is not masked.
31:6	Reserved		

5 System Configuration Controller (SYSCFG)

5.1 Full Name and Abbreviation Description of Terms

Table 16 Full Names and Abbreviations of Terms

Full Name	Abbreviation
Fast Mode Plus	FM+
System Configuration Controller	SYSCFG

5.2 Introduction

The SYSCFG (System Configuration Controller) is primarily used for managing memory remapping and controlling interrupt sources. Its key functions include:

- Configuring DMA trigger source remapping
- Remapping memory to the code start region
- Managing external interrupts connected to GPIOs

For detailed configuration information, refer to the SYSCFG Register Configuration.

5.3 Register Address Mapping

Table 17 SYSCFG Register Address Mapping

Register name	Description	Offset address
SYSCFG_CFG1	Configuration register 1	0x00
SYSCFG_EINTCFG1	External interrupt register 1	0x08
SYSCFG_EINTCFG2	External interrupt register 2	0x0C
SYSCFG_EINTCFG3	External interrupt register 3	0x10
SYSCFG_EINTCFG4	External interrupt register 4	0x14
SYSCFG_CFG2	Configuration register 2	0x18

5.4 Register Functional Description

5.4.1 Configuration Register 1 (SYSCFG_CFG1)

Offset address: 0x00

Reset value: 0x0000 000X (X means memory mode, controlled by nBOOTx. After reset, these bits select mode configuration parameters through nBOOTx.)

This register is used to configure the memory and DMA requested remapping and control the specific I/O pins.

Two bytes are used to configure the storage type mapped with the address of 0x0000 0000.

All of these bits can skip the hardware to have the software to select the physical mapping, and can be controlled and reset by software.

Field	Name	R/W	Description
1:0	MMSEL	R/W	Select Memory Mapping Control the memory mapping at address 0x0000 0000. The values of these bits after a reset are determined by the physical state of the nBOOT0/1 pins. x0: Main flash mapping address: 0x0000 0000 01: Reserved 11: Embedded SRAM mapping address: 0x0000 0000
7:2	Reserved		
8	ADCDMARMP	R/W	ADC DMA Request Remapping Control remapping request of ADC DMA. 0: No remapping ADC—DMA_CH1 1: Remapping ADC—DMA_CH2
9	USART1TXRMP	R/W	USART1_TX DMA Request Remapping This bit controls remapping request of USART1_TX DMA. 0: No remapping USART1_TX—DMA_CH2 1: Remapping USART1_TX—DMA_CH4
10	USART1RXRMP	R/W	USART1_RX DMA Request Remapping This bit controls remapping request of USART1_RX DMA. 0: No remapping USART1_RX—DMA_CH3 1: Remapping USART1_RX—DMA_CH5
15:11	Reserved		
16	TMR4CH3RMP	R/W	TMR4_CH3 DMA Request Remapping This bit controls remapping request of TMR4_CH3 DMA. 0: No remapping: TMR4_CH3—DMA_CH1 1: Remapping: TMR4_CH3—DMA_CH5
17	TMR4UPRMP	R/W	TMR4_UP DMA Request Remapping This bit controls remapping request of TMR4_UP DMA. 0: No remapping: TMR4_UP—DMA_CH1 1: Remapping: TMR4_UP—DMA_CH5
31:18	Reserved		

5.4.2 External Interrupt Register 1 (SYSCFG_EINTCFG1)

These bits are controlled by software to be rewritten to select the external interrupt source of EINTx (x=0…3). The selected external interrupt sources

represented by the values of the EINTx [3:0] are shown in the table below.

Table 18 External Interrupt Sources Selected for Different Values

EINTx [3:0]	External Interrupt Source
x000	PA[x] pin
x001	PB[x] pin
x010	PC[x] pin
x011	PD[x] pin
x100	Reserved
x101	PF[x] pin
Others	Reserved

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	EINT0	R/W	EINT0 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT0. The selected external interrupt sources represented by the values of different bits are shown in the table "External Interrupt Sources Selected for Different Values".
7:4	EINT1	R/W	EINT1 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT1. The selected external interrupt sources represented by the values of different bits are shown in the table "External Interrupt Sources Selected for Different Values".
11:8	EINT2	R/W	EINT2 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT2. The selected external interrupt sources represented by the values of different bits are shown in the table "External Interrupt Sources Selected for Different Values".
15:12	EINT3	R/W	EINT3 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT3. The selected external interrupt sources represented by the values of different bits are shown in the table "External Interrupt Sources Selected for Different Values".
31:16	Reserved		

5.4.3 External Interrupt Register 2 (SYSCFG_EINTCFG2)

These bits are controlled by software to be rewritten to select the external interrupt source of EINTx(x=4…7). The selected external interrupt sources represented by the values of EINTx [3:0] are shown in the table "External Interrupt Sources Selected for Different Values".

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	EINT4	R/W	EINT4 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT4. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
7:4	EINT5	R/W	EINT5 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT5. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
11:8	EINT6	R/W	EINT6 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT6. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
15:12	EINT7	R/W	EINT7 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT7. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
31:16	Reserved		

5.4.4 External Interrupt Register 3 (SYSCFG_EINTCFG3)

These bits are controlled by software to be rewritten to select the external interrupt source of EINTx(x=8…11). The selected external interrupt sources represented by the values of EINTx [3:0] are shown in the table “External Interrupt Sources Selected for Different Values”.

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	EINT8	R/W	EINT8 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT8. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.

Field	Name	R/W	Description
7:4	EINT9	R/W	EINT9 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT9. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
11:8	EINT10	R/W	EINT10 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT10. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
15:12	EINT11	R/W	EINT11 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT11. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
31:16	Reserved		

5.4.5 External Interrupt Register 4 (SYSCFG_EINTCFG4)

These bits are controlled by software to be rewritten to select the external interrupt source of EINTx(x=12 to 15). The selected external interrupt sources represented by the values of EINTx [3:0] are shown in the table “External Interrupt Sources Selected for Different Values”.

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	EINT12	R/W	EINT12 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT12. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
7:4	EINT13	R/W	EINT13 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT13. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
11:8	EINT14	R/W	EINT14 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT14. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.

Field	Name	R/W	Description
15:12	EINT15	R/W	EINT15 Configure These bits are controlled by software to be rewritten to select the external interrupt source of EINT15. The selected external interrupt sources represented by the values of different bits are shown in the table “External Interrupt Sources Selected for Different Values”.
31:16	Reserved		

5.4.6 Configuration Register 2 (SYSCFG_CFG2)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	LOCK	R/W	Core LOCKUP Enable This bit is set by software and cleared by system reset. It can enable and lock the connection between Arm® Cortex® -M0+ LOCKUP Hardfault (hardware error) output and TMR1 brake input.
1	Reserved		
2	PVDLOCK	R/W	PVD Lock Enable It can enable and lock the connection between PVD interrupt and TMR1 brake input and lock the state of PVDEN bit and PLSEL bit in PMU_CTRL. 0: Connection locked; PVDEN bit and PLSEL bit are editable 1: Connection enabled; PVDEN bit and PLSEL bit are read-only
31:3	Reserved		

6 Reset and Clock Management (RCM)

6.1 Full Name and Abbreviation Description of Terms

Table 19 Full Names and Abbreviations of Terms

Full Name	Abbreviation
Reset and Clock Management	RCM
Reset	RST
Power-On Reset	POR
Power-Down Reset	PDR
High Speed External Clock	HSECLK
High Speed Internal Clock	HSICLK
Low Speed Internal Clock	LSICLK
Phase Locked Loop	PLL
Main clock output	MCO
Calibrate	CAL
Trim	TRM
Wakeup	WUP
Automatic Wakeup	AWUP
Low Power	LPWR
Clock Security System	CSS
Non Maskable Interrupt	NMI
Clock Monitoring Module	CMU

6.2 Reset Functional Description

The supported reset is divided into two forms, namely, system reset and power reset.

6.2.1 System Reset

6.2.1.1 "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:

- Independent watchdog termination count (IWDT reset)

- Software reset (SW reset)
- Low-power management reset
- Load option byte reset
- Power reset
- Loss-of-Clock Reset (LOC Reset)

A system reset will occur when any of the above events occurs. Besides, the reset event source can be identified by viewing the reset flag bit in RCM_CSTS (control/status register). The LOC Reset can be enabled or disabled by configuring the RCM_INT2[SLOCEN] register bit.

When the system is reset, all registers except the registers in RCM_CSTS (control/state register) reset flag bit (refer to the Power Supply Control Block Diagram) will be reset to the reset state.

Software Reset

Software can be reset by setting SYSRESETREQ in Arm® Cortex® -M0+ 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 RSTSTDBY bit (in standby mode) or RSTSTOP bit (in stop mode) in user-selected byte is cleared, the system will be reset and not enter the standby or stop mode.

For more information about user-selected bytes, refer to the chapter of "Flash Memory".

Load option byte reset

The load byte reset is triggered by OBLOAD bit in FMC_CTRL2 register which is controlled by software.

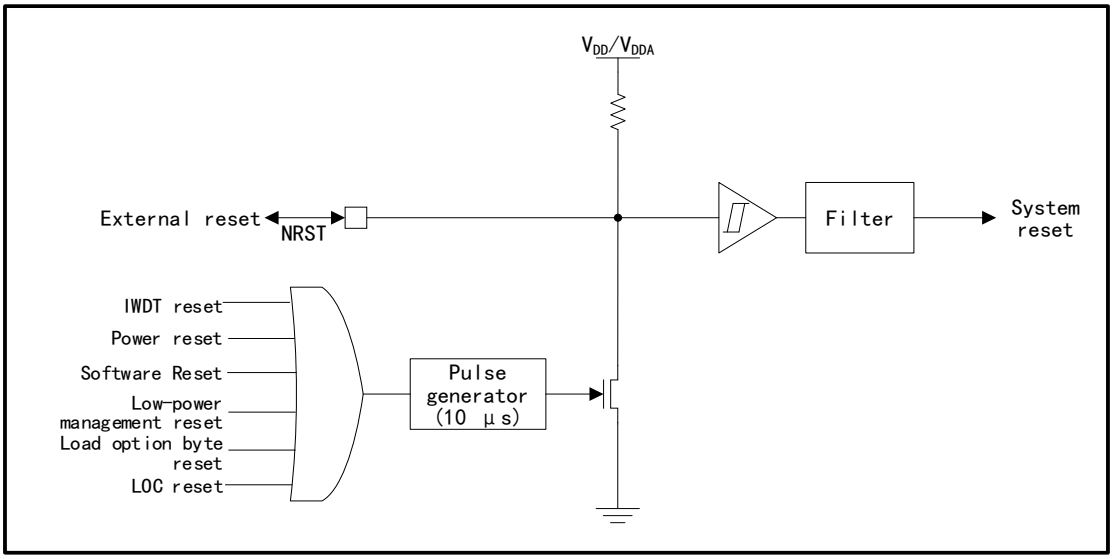
6.2.1.2 "System reset" reset circuit

All reset sources act on the NRST pin, which remains low during the reset process.

The internal reset source generates a pulse with a delay of at least 10 μs at the NRST pin through the pulse generator, which causes the NRST pin to maintain the level to generate reset; the external reset source directly pulls down the NRST pin level for resetting.

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

Figure 2 Reset Circuit for "System Reset"



6.2.2 Power Reset

"Power reset" reset source is as follows:

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

A power reset will occur when any of the above events occurs.

Power reset will reset all registers.

6.3 Functional Description of Clock Management

The clock sources of the whole system include HSECLK, HSICLK, HSICLK14, LSICLK and PLL. For the characteristics of the clock source, refer to the relevant chapter of "Electrical Characteristics" in the datasheet.

6.3.1 External Clock Source

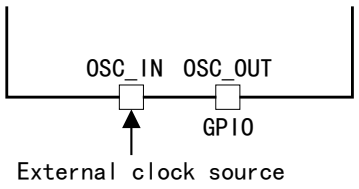
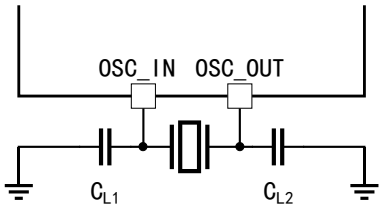
The external clock signal is HSECLK (high-speed external clock) signal.

There are two kinds of external clock sources:

- External crystal/ceramic oscillator
- External clock of user

The hardware configuration of HSECLK (high-speed external clock) signal is shown in the following figure.

Figure 3 HSECLK Hardware Configuration

Clock source	Hardware configuration
External clock	
Crystal/Ceramic resonator	

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

6.3.1.1 HSECLK (high-speed external clock) signal

HSECLK signal is generated by two clock sources, which are HSECLK external crystal/ceramic oscillator and HSECLK external clock.

Table 20 Clock Sources Generating HSECLK Signal

Name	Description
External clock source (HSECLK bypass)	Provide clock to the 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 40%-60% duty cycle, and the maximum frequency is up to 20MHz. In hardware connection, it should be connected to OSC_IN pin, ensuring OSC_OUT pin is suspended; in MCU configuration, the user can select this mode by setting HSEBCFG and HSEEN bits in RCM_CTRL1 (clock control register 1).

Name	Description
External crystal/ceramic oscillator (HSECLK crystal)	The clock is provided to MCU by the oscillator, and the oscillator includes crystal oscillator and ceramic oscillator. The frequency range is 8-20 MHz. When needing to connect OSC_IN and OSC_OUT to the oscillator, it can be enabled and disabled by setting the HSEEN bit in clock control register RCM_CTRL1 (clock control register). HSERDYFLG bit in the clock control register RCM_CTRL1 (clock control register 1) is used to indicate whether the high-speed external oscillator is stable. After it is enabled, 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.

6.3.2 Internal Clock Source

The internal clock sources include: HSICLK (High-Speed Internal Clock), HSICLK14 (High-Speed Internal 14 MHz Clock), and LSICLK (Low-Speed Internal Clock) signals.

6.3.2.1 HSICLK (High-speed internal clock) signal

HSICLK signal is generated by internal 8 MHz 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 frequency of each chip has been calibrated to 1% (25 ℃, LDO_CAP=3.3 V) by the manufacturer before leaving the factory. When the system is reset, the value calibrated by the manufacturer will be loaded to RCM_CTRL1 (clock control register); in addition, the users can further adjust the frequency by setting HSITRM in RCM_CTRL1 according to the application conditions (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 oscillator can be enabled or disabled by HSIEN bit in RCM_CTRL1.

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.

6.3.2.2 HSICLK14 (High-Speed Internal 14 MHz Clock) signal

HSICLK14 signal is generated by internal 14MHz 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 frequency of each chip has been calibrated to 1% (25 ℃, LDO_CAP=3.3 V) by

the manufacturer before leaving the factory. When the system is reset, the value calibrated by the manufacturer will be loaded to RCM_CTRL2 (clock control register); in addition, the users can further adjust the frequency by setting HSI14TRM in RCM_CTRL2 according to the application conditions (temperature and voltage) of the site.

HSI14RDYFLG bit can be used to indicate whether HSICLK14 RC oscillator is stable. In the clock startup process, HSICLK14 RC output clock is not released until the HSI14RDYFLG bit is set to "1" by hardware. HSICLK14 RC oscillator can be enabled or disabled by HSI14EN bit in RCM_CTRL2.

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.

6.3.2.3 LSICLK low-speed internal clock signal

Main Characteristics of LSICLK Signal

LSICLK signal is generated by RC oscillator at the frequency of 32 kHz. The frequency may change along with the change of temperature and voltage. It can keep running in both stop and standby mode and provide clock for IWDT (independent watchdog).

The LSICLK frequency of each chip has been calibrated by the manufacturer before leaving the factory. When the system is reset, the value calibrated by the manufacturer will be loaded to RCM_CTRL2 (clock control register); in addition, the users can further adjust the frequency by setting LSITRM in RCM_CTRL2 according to the application conditions (temperature and voltage) of the site.

LSICLK can be enabled or disabled by LSIEN bit of RCM_CSTS (control/status register). 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 it is allowed in RCM_INT (clock interrupt register), LSICLK interrupt request will be generated.

6.3.3 PLL (phase locked loop)

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

To configure PLL parameters, first clear PLLEN bit, and after PLLRDYFLG is cleared (PLL is in the disabled state), change the parameters, then set PLLEN to 1, enable PLL, and when PLLRDYFLG is set to 1, the configuration is completed. The clock source and multiplication factor should be selected before activated. Once PLL is activated, the selection cannot be changed.

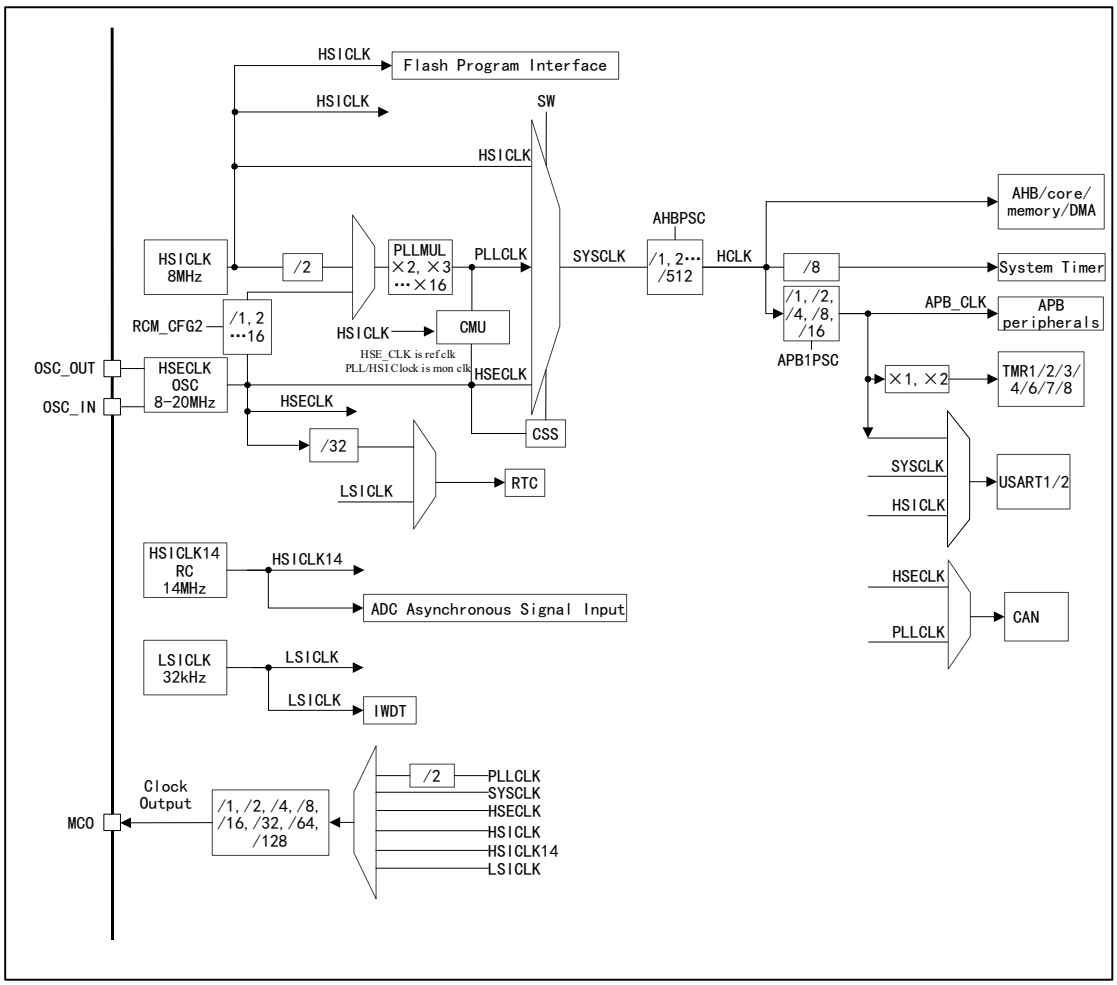
When PLL is ready and PLL interrupt in RCM_INT is allowed, PLL can generate

interrupt request.

6.3.4 Clock Tree

G32A1085 A1065 A1045 clock tree is shown in the figure below:

Figure 4 G32A1085 A1065 A1045 Clock Tree



Note:

(1) HCLK refers to an AHB clock, an operating clock (FCLK) of the Arm® Cortex®-M0+.

(2) PCLK is a clock signal of the peripheral connected to APB.

(3) The frequency of AHB and APB domains can be configured by multiple prescalers.

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

(5) The clock frequency for all TMRxCLK (Timer clocks) except TMR8 is automatically configured by hardware based on the following two conditions:

- If the corresponding APB prescaler factor is 1, the clock frequency of the timer is the same as the frequency 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.
- The clock frequency of the TMR8 timer is identical to that of APB bus connected to the timer.

(6) The frequency of TMRx (x=1, 2, 3, 4, 6, 7, 8) clock signals is divided by APB.

6.3.5 Clock Source Selection of IWDT

When IWDT (independent watchdog) is enabled, LSICLK oscillator will be enabled by force, and when it is stable, it will provide the clock signal to IWDT. After LSICLK is enabled by force, it will always be enabled and cannot be disabled.

6.3.6 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 configuring MCOSEL in RCM_CFG1 (clock configuration register). See the instructions for clock tree or MCOSEL bit of RCM_CFG1 register for specific clock signals. The available division factors for the selected clock source are 1, 2, 4, 8, 16, 32, 64, and 128. This configuration is specified by the MCOPRE field in the RCM_CFG1 register.

6.3.7 SYSCLK Source Selection

SYSCLK sources can be HSECLK, PLLCLK and HSICLK.

The status bit of RCM_CFG1 can indicate the ready clock and selected SYSCLK clock source.

After the system is reset, the clock source cannot be stopped if HSICLK oscillator is selected as the system clock, or PLL is directly or indirectly used as the system clock. If you want to switch the SYSCLK clock source, you must wait until the target clock source is ready, which means the target clock source is stable.

6.3.8 CSS Clock Security System

In order to prevent MCU from failing to to run normally due to short circuit of external crystal oscillator, MCU can activate CSS clock security system by 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 the system clock), the external HSECLK oscillator will be disabled 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 disabled.

CSS can be activated by software. When HSECLK clock fails, CSS interrupt will be generated, and NMI will be generated automatically. NMI will be executed

continuously until the CSS interrupt pending bit is cleared. Therefore, CSS interrupt must be cleared by setting CSSCLR bit of RCM_INT (clock interrupt register) in NMI processing program.

6.3.9 Clock Source Selection of ADC

The clock source of ADC is controlled by ADC_CFG2. It can select HSICLK14 or PCLK with the frequency divided by 2/4 as the clock source. When PCLK is used as the clock source of ADC, HSICLK14 cannot be changed over to ADC interface.

6.3.10 Low-Power Mode

The peripheral clock and DMA clock of APB can be disabled by software.

Sleep mode:

- Stop CPU clock
- Flash and RAM interface clocks can be stopped by software
- When all peripheral clocks connected to APB bus are disabled, the AHB1/APB bridge clocks can be stopped by hardware

Stop mode:

- PLLCLK, HSICLK, HSICLK14 and HSECLK are disabled

Standby mode:

- All 1.2V power domains are disabled
- PLLCLK, HSICLK, HSICLK14 and HSECLK are disabled

Deep sleep mode:

- The system can be debugged by setting STOP_CLK_STS bit and STANDBY_CLK_STS bit in DBGMCU_CFG.
- The system selects HSICLK as SYSCLK by interrupt (in stop mode) or reset (standby mode)
- If Flash programming is in progress, the system will enter the deep sleep mode only after all programming operations are completed
- If APB domain is being used, the system will enter the deep sleep mode only after all operations are completed

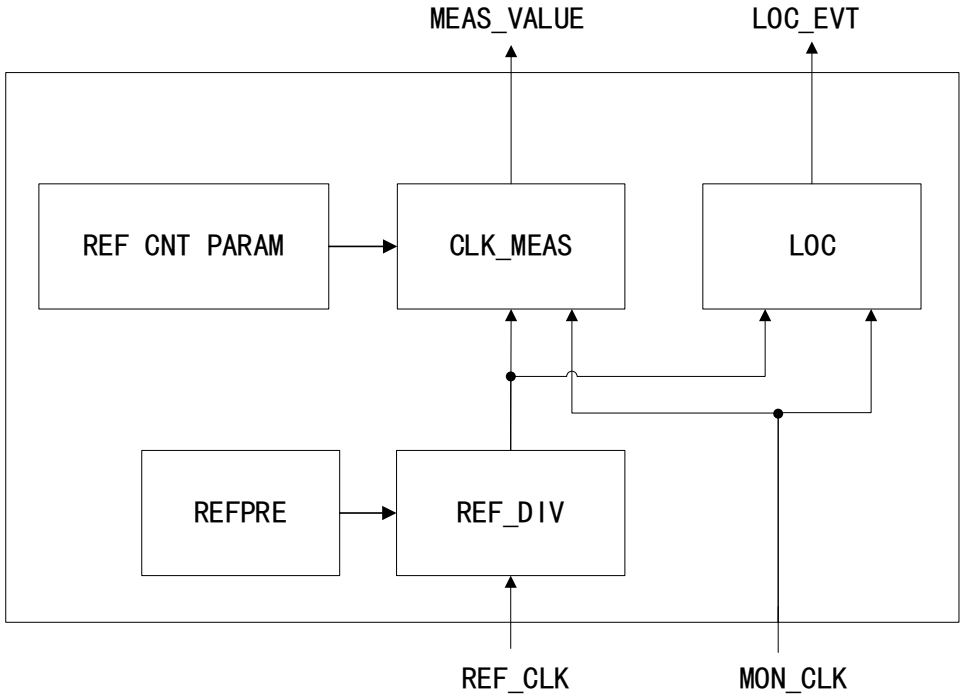
6.4 CMU Monitoring Function Description

The CMU (Clock Monitor Unit) circuit primarily consists of a Clock Measurement module (CLK_MEAS) and a Loss of Clock Detection module (CLK_LOC), designed to monitor whether the clock frequency is normal.

- Reference clock counting period can be configured.
- Monitors the clock frequency.
- Supports interrupt and interrupt masking.
- Clock Loss Monitoring.
- Supports system reset upon loss of clock.

6.4.1 Structure Block Diagram

Figure 5 CMU Structure Block Diagram



Note: REF_CLK is a reference clock, and MON_CLK is a monitoring clock.

6.4.2 Functional Description

The monitoring function for the measured clock is implemented by counting the cycles of the reference clock for a specific duration. During this counting period, the measured clock is counted synchronously. After the counting cycle ends, clock monitoring functionality is enabled via frequency measurement. The measured value is automatically updated into the measurement value register at the end of each cycle. During the counting process, the measured value remains latched to the previous measurement. It is required that the frequencies of both the reference clock and the measured clock remain stable during the counting period. If the clock frequency changes, it will affect the measurement result, rendering that particular measurement invalid. Furthermore, if the frequencies of the reference clock and the measured clock exceed the range specified in the datasheet, the measurement result may also be invalid.

6.4.3 Reference Clock

The reference clock is the HSE clock and used to monitor the status of the measured clock. Either HSICLK or PLL can be selected as the monitored clock. It is recommended to operate the CMU clock within the specified frequency range. If the frequency exceeds this range, the accuracy of the measurement results cannot be guaranteed. The CMU includes a reference clock divider which is used to configure the HSE frequency division, generating a 1 MHz

fractional frequency clock. For example, if the external crystal oscillator is 20 MHz, setting RCM_CFG4[REFPRE] to 19 will produce the 1 MHz clock.

6.4.4 Measurement Function

Once the CMU is enabled, it continuously monitors the frequency of the monitored clock, performing periodic measurements and recording the clock frequency. The measurement result is automatically updated upon completion of each measurement cycle. This result value can be obtained by reading the RCM_MEAS register. The measurement window can be configured via the RCM_REF_CNT[REF_CNT] register. The measurement period is determined by the following formula:

$$T_{MEAS} = (N_{REFPRE} + 1) * (N_{REF\_CNT} + 1) / f_{REF}$$

Wherein, N_{REF_CNT} is the value configured in the RCM_REF_CNT [REF_CNT] register; N_{REFPRE} is the value configured in the RCM_CFG4 [REFPRE] register; f_{REF} is the reference clock frequency. For example, if f_{REF}= 20 MHz, N_{REF_CNT}= 0x3F, and N_{REFPRE}= 0x1F, the resulting measurement period is 102.4 μs. Note that if the reference clock frequency is very low and the RCM_REF_CNT [REF_CNT] value is set very high, the measurement period can become relatively long.

When the frequencies of both the reference clock and the measured clock are within the specified design range, the expected reference value for the measurement result can be calculated using the following formula:

$$N_{MEAS\_VAL,ref} = ((N_{REFPRE} + 1) * f_{MON} * (N_{REF_{CNT}} + 1) / f_{REF}) - 1$$

Wherein, N_{REF_CNT} is the value configured in the RCM_REF_CNT [REF_CNT] register; f_{MON} is the monitored clock frequency; $N_{REFPRE}$is the value configured in the RCM_CFG4 [REFPRE] register; f_{REF} is the reference clock frequency.

For instance, if f_{REF} = 16 MHz, f_{MON} = 64 MHz (PLL), with configurations N_{REFPRE}= 0xF and N_{REF_CNT}= 0x13, the measurement period would be 20 μs, and the expected measurement result reference value would be 0x4FF.

To avoid measurement errors, it is necessary to read the measurement results multiple times consecutively. A measurement result is considered valid only when two consecutive readings are identical. When measuring frequency, it is recommended that the AHB clock is not divided (no frequency division) to ensure that multiple readings can be obtained within a single measurement window.

6.4.5 Measurement Accuracy

The prerequisite for CMU monitoring is that the reference clock frequency is known and stable. The measurement deviation is within two measured clock cycles: one from the clock edge capture error and the other from the measurement clock counter error. Measurement accuracy is also affected by the

measurement period and clock precision. Higher measured clock frequencies and longer measurement periods result in relatively higher measurement accuracy.

6.4.6 Loss of Clock Detection

After the CMU is enabled, it will monitor the measured clock for loss. If a clock loss is detected, a LOC (Loss-of-Clock) event is generated. The status of a LOC event can be determined by reading RCM_CFG4 [10]. The detection time window for a lost clock is approximately 10 μs to 30 μs. The CMU cannot detect the loss of the reference clock. If the reference clock is lost, the monitoring results become invalid. A measured clock frequency below 1 MHz is considered a clock loss, causing the system to generate a LOC event. The priority of the LOC monitoring result is higher than that of the frequency measurement result. When the measured clock frequency is less than or equal to 1 MHz, the frequency measurement result may be unreliable. It is recommended to prioritize the LOC monitoring result for judgement, followed by the frequency measurement result.

6.4.7 Interrupt and Reset

If the LOCIE bit is enabled, an RCM interrupt will be generated upon the occurrence of LOC events. If the SLOC bit is enabled, a system reset will be implemented upon a LOC event. The RCM_CSTS [22] bit indicates a LOC reset source.

6.4.8 Low-Power Mode

When entering STOP mode, the CMU is automatically disabled by hardware. After wake-up, the HSE clock must be re-enabled. The measurement result becomes valid only after the VLD flag is set to 1. If the measured clock is the PLL clock, the PLL clock must also be re-enabled, and measurements should only be performed after the PLLRDY flag becomes valid.

6.5 Register Address Mapping

Table 21 RCM Register Address Mapping

Register name	Description	Offset address
RCM_CTRL1	Clock Control Register 1	0x00
RCM_CFG1	Clock Configuration Register 1	0x04
RCM_INT1	Clock Interrupt Register	0x08
RCM_APBRST2	APB Peripheral Reset Register 2	0x0C
RCM_APBRST1	APB Peripheral Reset Register 1	0x10
RCM_AHBCLKEN	AHB peripheral clock enable register	0x14

Register name	Description	Offset address
RCM_APBCLKEN2	APB Peripheral Clock Enable Register 2	0x18
RCM_APBCLKEN1	APB Peripheral Clock Enable Register 1	0x1C
RCM_RTCCTRL	RTC Control Register	0x20
RCM_CSTS	Control/Status Register	0x24
RCM_AHBRST	AHB peripheral reset register	0x28
RCM_CFG2	Clock Configuration Register 2	0x2C
RCM_CFG3	Clock Configuration Register 3	0x30
RCM_CTRL2	Clock Control Register 2	0x34
RCM_INT2	Clock Interrupt Register 2	0x40
RCM_CFG4	Clock Configuration Register 4	0x44
RCM_REF_CNT	Clock Cycle Configuration Register	0x48
RCM_MEAS	Measurement Result Latch Register	0x4C

6.6 Register Functional Description

6.6.1 Clock Control Register 1 (RCM_CTRL1)

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	Enable High Speed Internal Clock Set to 1 or cleared by software. HSICLK is an RC oscillator. When one of the following conditions occurs, it will be set to 1 by 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 enabled
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[4:0].

Field	Name	R/W	Description
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 disabled; 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
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 oscillator 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 to0 by 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 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		

6.6.2 Clock Configuration Register 1 (RCM_CFG1)

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	Select System Clock Source When returning from Stop or Standby mode, or when a failure occurs in HSECLK (either directly or indirectly serving as the system clock), the hardware will select HSICLK as the system clock (if the clock security system has been enabled). 00: HSICLK is used as system clock 01: HSECLK is used as system clock 10: PLLCLK is used as system clock 11: Reserved
3:2	SCLKSELSTS	R	System Clock Selection Status Indicate which clock source is used as system clock; set to 1 or clear to 0 by hardware. 00: HSICLK is used as system clock 01: HSECLK is used as system clock 10: PLLCLK is used as system clock 11: Reserved
7:4	AHBPSC	R/W	AHB Clock Prescaler Factor Configure Control the prescaler factor of AHB clock. 0xxx: No division 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 the AHB clock is greater than 1, the prefetch buffer must be enabled.
10:8	APB1PSC	R/W	Configure APB1 Clock Prescaler Factor Control the prescaler factor of low-speed APB1 clock (PCLK1) 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
15:11	Reserved		
16	PLLSRCSEL	R/W	PLL Clock Source Select This bit can be changed only when PLL is disabled. 0: HSICLK oscillator is used as PLL clock source after 2 frequency division 1: HSECLK is used as PLL clock source after frequency division
17	PLLHSEPSC	R/W	HSECLK Prescaler Factor for PLL Clock Source Refer to Bit 0 of RCM_CFG2.

Field	Name	R/W	Description
21:18	PLLMULCFG	R/W	PLL Multiplication Factor Configure Determine PLL multiplication factor. This bit can be written only when PLL is disabled. 0000: PLL 2-multiple frequency output 0001: PLL 3-multiple frequency output 0010: PLL 4-multiple frequency output 0011: PLL 5-multiple frequency output 0100: PLL 6-multiple frequency output 0101: PLL 7-multiple frequency output 0110: PLL 8-multiple frequency output 0111: PLL 9-multiple frequency output 1000: PLL 10-multiple frequency output 1001: PLL 11-multiple frequency output 1010: PLL 12-multiple frequency output 1011: PLL 13-multiple frequency output 1100: PLL 14-multiple frequency output 1101: PLL 15-multiple frequency output 1110: PLL 16-multiple frequency output 1111: PLL 16-multiple frequency output Note: The output frequency of PLL cannot be greater than 72MHz.
23:22	Reserved		
27:24	MCOSEL	R/W	Main Clock Output Select Set or clear 0 by software. 0000: No clock output 0001: HSICLK14 is output as a clock 0010: LSICLK is output as a clock 0011: Reserved 0100: SYSCLK is output as a clock 0101: HSICLK is output as a clock 0110: HSECLK is output as a clock 0111: PLLCLK is output as a clock (the divider factor is determined by MCOPLLPSC) 1000: Reserved
30:28	MCOPRE	R/W	Main Clock Output Prescaler Factor Used to configure the prescaler for the MCO (Microcontroller Clock Output). Modifying this prescaler factor may generate glitches. It is highly recommended to change the prescaler factor only after a reset and before enabling the external oscillator and PLL. 000: No division 001: 2 divided frequency 010: 4 divided frequency 011: 8 divided frequency 100: 16 divided frequency 101: 32 divided frequency 110: 64 divided frequency 111: 128 divided frequency
31	Reserved		

6.6.3 Clock Interrupt Register (RCM_INT1)

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 LSICLK is stable and LSIRDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by LSIRDYCLR. 0: No LSICLK ready interrupt 1: LSICLK ready interrupt occurred
1	Reserved		
2	HSIRDYFLG	R	HSICLK Ready Interrupt Flag When HSICLK is stable and HSIRDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by HSIRDYCLR. 0: No HSICLK ready interrupt 1: HSICLK ready interrupt occurred
3	HSERDYFLG	R	HSECLK Ready Interrupt Flag When HSECLK is stable and HSERDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by HSERDYCLR. 0: No HSECLK ready interrupt 1: HSECLK ready interrupt occurred
4	PLLRDYFLG	R	PLL Ready Interrupt Flag When PLL is stable and PLLRDYEN bit is set to 1, set 1 by hardware; set 1 by software and clear 0 by PLLRDYCLR. 0: No clock ready interrupt caused by PLL locked 1: Clock ready interrupt caused by PLL locked
5	HSI14RDYFLG	R	HSICLK14 Ready Interrupt Flag When the internal high-speed clock is ready and HSI14RDYEN bit is set to 1, set 1 by hardware. Set 1 by software and clear 0 by HSI14RDYCLR. 0: HSICLK14 not ready 1: HSICLK14 ready
6	Reserved		
7	CSSFLG	R	Clock Security System Interrupt Flag When the external oscillator clock fails, it is set to 1 by hardware. This bit is set to "1" by software, and cleared by CSSCLR. 0: No security system interrupt caused by HSECLK failure 1: Clock security system interrupt caused by HSECLK failure
8	LSIRDYEN	R/W	Enable LSICLK Ready Interrupt Enable or disable internal RC oscillator ready interrupt. 0: Disable 1: Enable
9	Reserved		

Field	Name	R/W	Description
10	HSIRDYEN	R/W	HSICLK Ready Interrupt Enable Enable internal 8MHz RC oscillator ready interrupt. 0: Disable 1: Enable
11	HSERDYEN	R/W	Enable HSECLK Ready Interrupt Enable external oscillator ready interrupt. 0: Disable 1: Enable
12	PLLRDYEN	R/W	PLL Ready Interrupt Enable Enable PLL ready interrupt. 0: Disable 1: Enable
13	HSI14RDYEN	R/W	HSICLK14 Ready Interrupt Enable Enable internal 14MHz RC oscillator ready interrupt. 0: Disable 1: Enable
15:14	Reserved		
16	LSIRDYCLR	W	LSICLK Ready Interrupt Clear Clear LSICLK ready interrupt flag bit LSIRDYFLG. 0: No effect 1: Clear
17	Reserved		
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
20	PLLRDYCLR	W	PLL Ready Interrupt Clear Clear PLL ready interrupt flag bit PLLRDYFLG. 0: No effect 1: Clear
21	HSI14RDYCLR	W	HSICLK14 Ready Interrupt Clear Clear HSICLK14 ready interrupt flag bit HSI14RDYFLG. 0: No effect 1: Clear
22	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		

6.6.4 APB Peripheral Reset Register 2 (RCM_APBRST2)

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 set or cleared by software.

Field	Name	R/W	Description
0	SYSCFGRST	R/W	SYSCFG Reset 0: No effect 1: Reset
8:1	Reserved		
9	ADCRST	R/W	ADC Reset 0: No effect 1: Reset ADC
10	Reserved		
11	TMR1RST	R/W	TMR1 Timer Reset 0: No effect 1: Reset
12	SPI1RST	R/W	Reset SPI1 (SPI1 Reset) 0: No effect 1: Reset
13	Reserved		
14	USART1RST	R/W	USART1 Reset 0: No effect 1: Reset
15	TMR8RST	R/W	TMR8 Timer Reset 0: No effect 1: Reset
21:16	Reserved		
22	DBGRST	R/W	Reset Debug 0: No effect 1: Reset
23	SMSRST	R/W	Reset SMS 0: No effect 1: Reset
31:24	Reserved		

6.6.5 APB Peripheral Reset Register 1 (RCM_APBRST1)

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	TMR2RST	R/W	Timer 2 Reset 0: No effect 1: Reset

Field	Name	R/W	Description
1	TMR3RST	R/W	Timer 3 Reset 0: No effect 1: Reset
2	TMR4RST	R/W	Timer 4 Reset 0: No effect 1: Reset
3	Reserved		
4	TMR6RST	R/W	Timer 6 Reset 0: No effect 1: Reset
5	TMR7RST	R/W	Timer 7 Reset 0: No effect 1: Reset
10:6	Reserved		
11	WWDTRST	R/W	Window Watchdog Reset 0: No effect 1: Reset
16:12	Reserved		
17	USART2RST	R/W	USART2 Reset 0: No effect 1: Reset
24:18	Reserved		
25	CANRST	R/W	CAN Reset 0: No effect 1: Reset
27:26	Reserved		
28	PMURST	R/W	Power Interface Reset 0: No effect 1: Reset
31:29	Reserved		

6.6.6 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 set 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	DMA1EN	R/W	DMA1 Clock Enable 0: Disable 1: Enable

Field	Name	R/W	Description
1	Reserved		
2	SRAMEN	R/W	SRAM Interface Clock Enable Enable SRAM clock in sleep mode. 0: Disable 1: Enable
3	Reserved		
4	FMCEN	R/W	FMC Clock Enable Enable the flash interface circuit clock in sleep mode. 0: Disable 1: Enable
5	Reserved		
6	CRCEN	R/W	CRC Clock Enable 0: Disable 1: Enable
7	TRNGEN	R/W	TRNG Clock Enable 0: Disable 1: Enable
8	SHAEN	R/W	SHA Clock Enable 0: Disable 1: Enable
9	AES256EN	R/W	AES256 Clock Enable 0: Disable 1: Enable
16:10	Reserved		
17	PAEN	R/W	I/O PortA Clock Enable 0: Disable 1: Enable
18	PBEN	R/W	I/O PortB Clock Enable 0: Disable 1: Enable
19	PCEN	R/W	I/O PortC Clock Enable 0: Disable 1: Enable
20	PDEN	R/W	I/O Port D Clock Enable 0: Disable 1: Enable
21	Reserved		
22	PFEN	R/W	I/O PortF Clock Enable 0: Disable 1: Enable
31:23	Reserved		

6.6.7 APB Peripheral Clock Enable Register 2 (RCM_APBCLKEN2)

Offset address: 0x18

Reset value: 0x0080 0000

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

Usually there is no access 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	SCFGEN	R/W	SYSCFG Clock Enable 0: Disable 1: Enable
8:1	Reserved		
9	ADCEN	R/W	ADC Interface Clock Enable 0: Disable 1: Enable
10	Reserved		
11	TMR1EN	R/W	TMR1 Timer Clock Enable 0: Disable 1: Enable
12	SPIEN	R/W	SPI Clock Enable 0: Disable 1: Enable
13	Reserved		
14	USART1EN	R/W	USART1 Clock Enable 0: Disable 1: Enable
15	TMR8EN	R/W	TMR8 Timer Clock Enable 0: Disable 1: Enable
21:16	Reserved		
22	DBGEN	R/W	Debug Clock Enable 0: Disable 1: Enable
23	SMSEN	R/W	SMS Clock Enable 0: Disable 1: Enable (reset value)
31:24	Reserved		

6.6.8 APB Peripheral Clock Enable Register 1 (RCM_APBCLKEN1)

Offset address: 0x1C

Reset value: 0x0000 0000

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

Usually there is no access wait cycle. However, when the peripheral on the APB bus is accessed, the waiting state will be inserted until the APB 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	TMR2EN	R/W	Timer2 Clock Enable 0: Disable 1: Enable
1	TMR3EN	R/W	Timer 3 Clock Enable 0: Disable 1: Enable
2	TMR4EN	R/W	Timer 4 Clock Enable 0: Disable 1: Enable
3	Reserved		
4	TMR6EN	R/W	Timer 6 Clock Enable 0: Disable 1: Enable
5	TMR7EN	R/W	Timer 7 Clock Enable 0: Disable 1: Enable
16:6	Reserved		
17	USART2EN	R/W	USART 2 Clock Enable 0: Disable 1: Enable
24:18	Reserved		
25	CANEN	R/W	CAN Clock Enable 0: Disable 1: Enable
27:26	Reserved		
28	PMUEN	R/W	Power Interface Clock Enable 0: Disable 1: Enable
31:29	Reserved		

6.6.9 RTC Control Register (RCM_RTCCTRL)

Offset address: 0x20

Reset value: 0x0000 0000

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 RTCSRCSEL and RTCCLKEN be changed.

Field	Name	R/W	Description
7:0	Reserved		
9:8	RTCSRCSEL	R/W	RTC Source Select First set the RTCRST bit to reset the RTC, and then select the RTC source. It is impossible to directly configure the register to modify. 00: No clock 01: Reserved 10: LSICLK is used as RTC clock 11: HSECLK is used as RTC clock after the 32-divide
14:10	Reserved		
15	RTCCLKEN	R/W	RTC Clock Enable 0: Disable 1: Enable
16	RTCRST	R/W	RTC Reset This function is only applicable for RTC and its controller. 0: No effect 1: Reset
31:17	Reserved		

6.6.10 Control/Status Register (RCM_CSTS)

Offset address: 0x24

Reset value: 0xXXX0 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 to 1 or cleared by software. 0: Disable 1: Enable
1	LSIRDYFLG	R	Low-Speed Internal Oscillator Ready Flag Set 1 by hardware when LSICLK is stable, and clear 0 by hardware when it is unstable. 0: Not ready 1. Ready
21:2	Reserved		
22	LOCRSTFLG	R	LOC Reset Flag Cleared by setting RSTFLGCLR. 0: No reset 1: Reset

Field	Name	R/W	Description
23	PWRRSTFLG	R	Reset Flag of 1.2 V Domain It is set to "1" by hardware when the 1.2 V power domain is set, and cleared by setting the RSTFLGCLR bit. 0: No reset 1: Reset
24	RSTFLGCLR	RT_W	Reset Flag Clear Set or clear reset flag by software, including RSTFLGCLR. 0: No effect 1: Clear reset flag
25	OBRSTFLG	R	Option Byte Loader Reset Flag When the option byte load reset occurs, set by hardware; otherwise, clear by setting RSTFLGCLR. 0: No reset 1: Reset
26	PINRSTFLG	R	PIN Reset Flag Set by hardware when pin reset occurs; otherwise, clear by setting RSTFLGCLR. 0: No reset 1: Reset
27	PODRSTFLG	R	POR/PDR Reset Occur Flag 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	Software Reset Occur Flag Set to "1" by hardware; and cleared by software by writing RSTFLGCLR bit. 0: Not occur 1: Occurred
29	IWDTRSTFLG	R	Independent Watchdog Reset Flag Set to 1 by hardware when independent watchdog reset occurs in V_{DD} area; clear by software by writing RSTFLGCLR bit. 0: Not occur 1: Occurred
30	Reserved		
31	LPWRRSTFLG	R	Low Power Reset Occur Flag Set 1 by hardware when low-power management is reset; clear by software by writing RSTFLGCLR bit. 0: Not occur 1: Occurred

6.6.11 AHB Peripheral Reset Register (RCM_AHBRST)

Offset address: 0x28

Reset value: 0x0000 0000

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

Set to 1 or cleared by software.

Field	Name	R/W	Description
6:0	Reserved		
7	TRNGEN	R/W	Reset TRNG Clock 0: Invalid 1: Reset
8	SHAEN	R/W	Reset SHA Clock 0: Invalid 1: Reset
9	AES256EN	R/W	Reset AES256 Clock 0: Invalid 1: Reset
16:10	Reserved		
17	PARST	R/W	I/O Port A Reset 0: Invalid 1: Reset
18	PBRST	R/W	I/O Port B Reset 0: Invalid 1: Reset
19	PCRST	R/W	I/O Port C Reset 0: Invalid 1: Reset
20	PDRST	R/W	Reset I/O Port D 0: Invalid 1: Reset
21	Reserved		
22	PFRST	R/W	I/O Port F Reset 0: Invalid 1: Reset
31:23	Reserved		

6.6.12 Clock Configuration Register 2 (RCM_CFG2)

Offset address: 0x2C

Reset value: 0x0000 0000

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

Field	Name	R/W	Description
3:0	PLLDIVCFG	R/W	Configure PLLCLK Input Division Factor Configure the input clock signal division factor of PLLCLK. 0000: No division 0001: 2 divided frequency 0010: 3 divided frequency .... 1111: 16 divided frequency
31:4	Reserved		

6.6.13 Clock Configuration Register 3 (RCM_CFG3)

Offset address: 0x30

Reset value: 0x0000 0000

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

Field	Name	R/W	Description
1:0	USART1SEL	R/W	USRAT1 Clock Source Select Set or cleared to 0 by software. The default value is 00. 00: PCLK is used as USART1CLK 01: SYSCLK is used as USART1CLK 10: Reserved 11: HSICLK is used as USART1CLK
3:2	USART2SEL	R/W	USRAT2 Clock Source Select Set or cleared to 0 by software. The default value is 00. 00: PCLK served as USART2CLK 01: SYSCLK served as USART2CLK 10: Reserved 11: HSICLK served as USART2CLK
11:4	Reserved		
12	CANSEL	R/W	CAN Clock Source Select Set or cleared to 0 by software. The default value is 00. 0: HSECLK served as CANCLK 1: PLLCLK served as CANCLK
31:13	Reserved		

6.6.14 Clock Control Register 2 (RCM_CTRL2)

Offset address: 0x34

Reset value: 0xXXA4 XX80; X means undefined

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

Field	Name	R/W	Description
0	HSI14EN	R/W	HSICLK14 Enable Set to 1 or cleared by software. 0: Internal 14MHz oscillator disabled 1: Internal 14MHz oscillator enabled
1	HSI14RDFLG	R	HSICLK14 Ready Flag Set by hardware, indicating the state of HSICLK14 oscillator. 0: Not ready 1: Ready
2	HSI14TO	R/W	ADC Interface Turn On HSICLK14 ADC interface can enable HSICLK14 oscillator; set or clear 0 by hardware. 0: Can enable 1: Cannot enable

Field	Name	R/W	Description
7:3	HSI14TRM	R/W	HSICLK14 Trim The product has been calibrated to 14MHz±1% when leaving the factory. However, it changes as the temperature and voltage changes, but the frequency of HSICLK14 RC oscillator can be adjusted by HSI14TRM.
15:8	HSI14CAL	R	HSICLK14 Calibrate It will be calibrated to 14 MHz±1% before leaving the factory. When the system is started up, the calibration parameters will be automatically written to the register.
17:16	Reserved		
23:18	LSITRM	R/W	LSI Trim The product is calibrated during factory testing, but the frequency may vary with temperature and voltage changes. The LSI frequency can be trimmed via the LSITRM register. Default value: 6b101001
31:24	LSICAL	R	LSI Calibration It will be calibrated before delivery. When the system is started up, the calibration parameters will be automatically written to the register. The default value is determined by the loaded value.

6.6.15 Clock Interrupt Register 2 (RCM_INT2)

Offset address: 0x40

Reset value: 0x0000 0000

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

Field	Name	R/W	Description
1:0	Reserved		
2	LOCFLG	R	Reference Clock Loss Interrupt Flag When LOCIE is enabled, this bit is set by hardware after an LOC event occurs. 0: No interrupt 1: Interrupt
9:3	Reserved		
10	LOCIE	R/W	Enable Reference Clock Loss Interrupt 0: Disable 1: Enable
17:11	Reserved		
18	LOCCLR	W	Clear the reference clock loss interrupt Clear LOCFLG 0: No effect 1: Clear
25:19	Reserved		
26	SLOCEN	R/W	Enable LOC Reset 0: No reset up on LOC event 1: Reset up on LOC event

Field	Name	R/W	Description
31:27	Reserved		

6.6.16 Clock Configuration Register 4 (RCM_CFG4)

Offset address: 0x44

Reset value: 0x0000 0000

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

Field	Name	R/W	Description
0	CMU_EN	R/W	Enable CMU 0: Disable 1: Enable
1	CLK_SEL	R/W	Select Measured Clock 0: HSI 1: PLL
9:2	Reserved		
10	LOC	RC_W1	Reference Clock Loss Event 0: Normal clock 1: Clock lost Write "1" to this bit to clear it. This reset is controlled by the Power-On Reset (POR) and cannot be triggered by a system reset.
11	VLD	R	Initial Frequency Measurement Valid Bit It indicates the end of initial measurement. 0: Frequency measurements invalid 1: Frequency measurements valid
15:12	Reserved		
21:16	REFPRE	R/W	Reference Clock Frequency Division Generate 1 MHz clock based on HSECLK frequency configuration 0: 1 divided frequency 1: 2 divided frequency 2: 3 divided frequency 3: 4 divided frequency … 49: 50 divided frequency 50~63: Reserved
31:22	Reserved		

6.6.17 Clock Cycle Configuration Register (RCM_REF_CNT)

Offset address: 0x48

Reset value: 0x0000 0000

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

Field	Name	R/W	Description
5:0	REF_CNT	R/W	Reference Clock Counting Cycle Measurement cycle: REF_CNT+1 It is recommended that the measurement cycle configuration value be greater than 15.

Field	Name	R/W	Description
31:6	Reserved		

6.6.18 Measurement Result Latch Register (RCM_MEAS)

Offset address: 0x4C

Reset value: 0x0000 0000

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

Field	Name	R/W	Description
15:0	MEAS_VAL	R	Measurement Result Latch Register
31:16	Reserved		

7 Power management unit (PMU)

7.1 Full Name and Abbreviation Description of Terms

Table 22 Full Names and Abbreviations of Terms

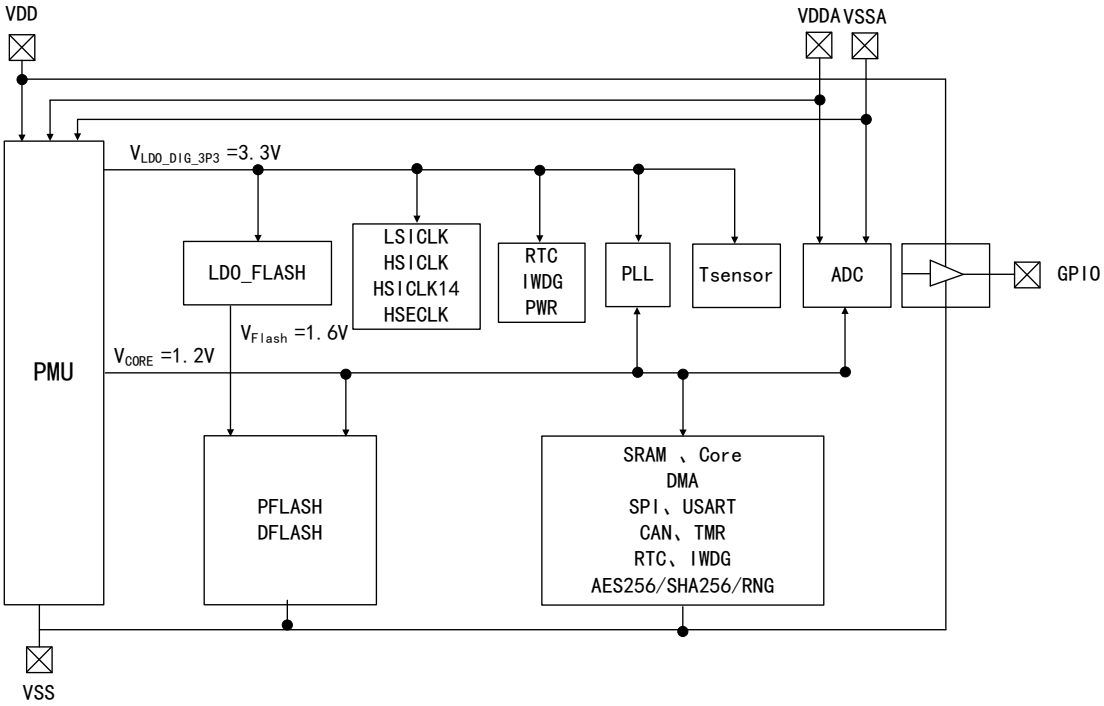
Full Name	Abbreviation
Power Management Unit	PMU
Power On Reset	POR
Power Down Reset	PDR
Power Voltage Detector	PVD

7.2 Introduction

The power supply is the foundation for stable operation of a system, with an operating voltage of 2.75- 5.5V, and 3.3 V, 1.6 V and 1.2 V power supply can be provided by the built-in voltage regulator.

7.3 Structure Block Diagram

Figure 6 Power Supply Control Structure Block Diagram



7.4 Functional Description

7.4.1 Power domain

The product includes the following power domains:V_{DD} power domain, V_{DDA} power domain, 3.3 V power domain, 1.6 V power domain and 1.2 V power domain.

7.4.1.1 V_{DD} power domain

It powers 3.3 V voltage regulator and I/O port via VDD /VSS pin.

3.3 V voltage regulator

It powers the 3.3 V power domain, including standby circuit, IWDT (Independent Watchdog Timer), wake-up logic and clock sources (HSI8M, HSI14M, HSE, LSI, PLL and Tsensor).

1.2 V voltage regulator

It powers the 1.2 V power domain in the following operating modes:

- Normal mode: In this mode, 1.2 V power domain runs at full power
- Stop mode: In this mode, 1.2 V power domain works in low-power state, all clocks are off, peripherals stop working and 3.3 V LDO enters the low-power mode.
- Standby mode: In this mode, the 1.2 V power domain is powered off, resulting in loss of 1.2 V domain register contents and SRAM data; the 3.3 V LDO switches to low-power mode.

1.6 V voltage regulator

It powers PFLASH and DFLASH.

7.4.1.2 V_{DDA} power domain

It powers PMU and ADC reset modules via VDDA /VSSA pin.

Independent ADC power supply

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

- VDDA : Power pin of ADC
- VSSA: Independent power ground pin

7.4.1.3 1.2V power domain

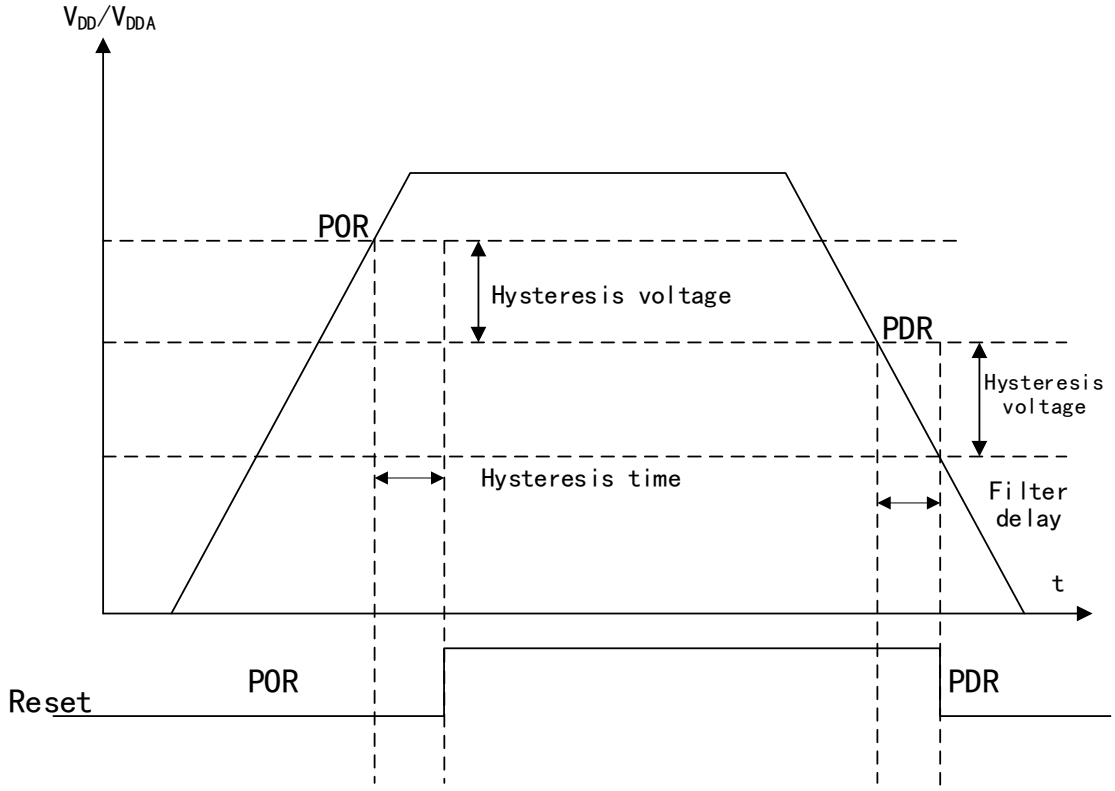
The core, Flash, SRAM and digital peripherals are powered by voltage regulator.

7.4.2 Power management

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

When the V_{DD} /V_{DDA} is lower than the threshold voltage V_{POR} and V_{PDR}, the chip will automatically remain in the reset state. The waveform diagram of power-on reset and power-down reset are shown in the figure below. For POR, PDR, hysteresis voltage, hysteresis time and filter delay, refer to the Datasheet.

Figure 7 Power-on Reset and Power-down Reset Waveform Diagram



7.4.2.2 Programmable 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 the interrupt is enabled, the interrupt can be triggered to process the V_{DD}/V_{DDA} 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 through the PLSEL[2:0] bit of the configuration register PMU_CTRL.

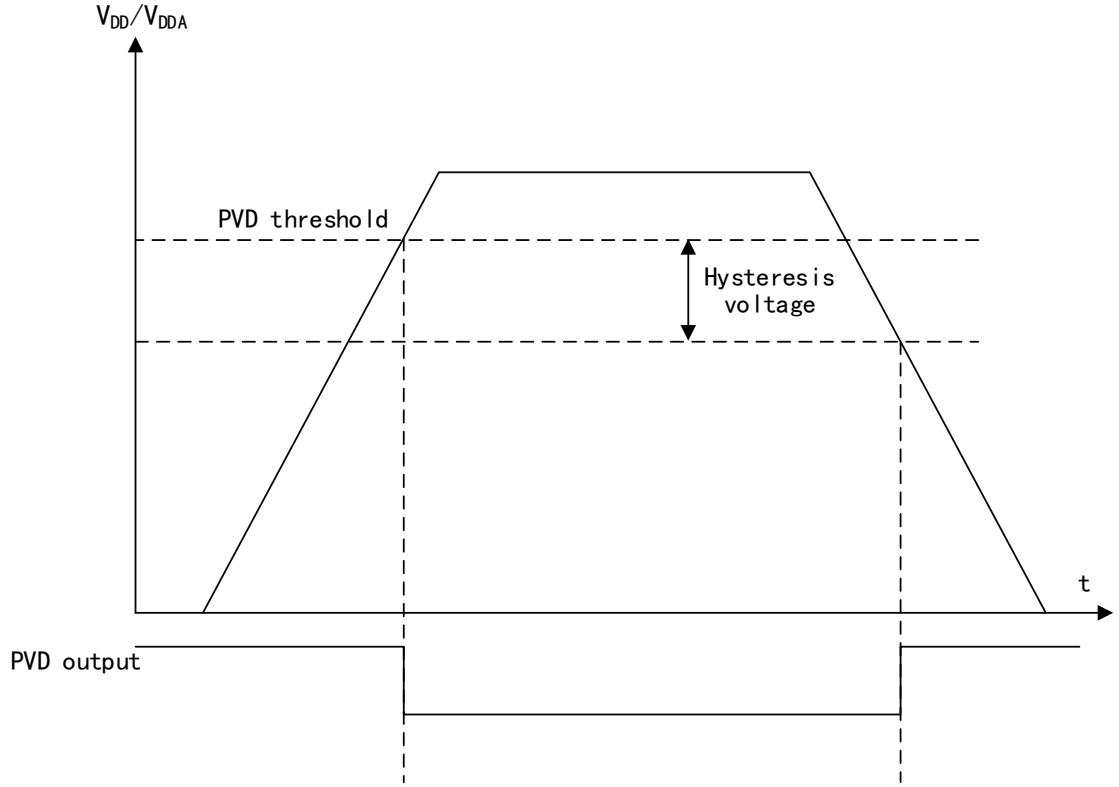
(3) The PVDOFLG bit of the configuration register PMU_CSTS indicates the value of V_{DD} higher or lower than the threshold of PVD.

(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 oscillogram of PVD is shown in the figure below. Refer to

"Datasheet" for PVD threshold and hysteresis voltage in details.

Figure 8 PVD Threshold Waveform Diagram



7.4.3 Power control

7.4.3.1 Reduce the power in low-power mode

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

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

Table 23 Difference among "SLEEP Mode and STOP Mode"

Mode	Description	Entry method	Wake-up mode	Voltage regulator	Effect on 1.2 V power domain clock	Effect on V_{DD} power domain clock
Sleep	Arm® Cortex®-M0+ core stops,	Call WFI instruction	Any interrupt	On	Only the core clock is	None

Mode	Description	Entry method	Wake-up mode	Voltage regulator	Effect on 1.2 V power domain clock	Effect on V_{DD} power domain clock
	and all peripherals including the core peripheral are still working	Call WFE instruction	Interrupt or wake-up event		disabled, and it has no effect on other clocks and ADC clocks	None
Stop	All clocks have stopped	PDDSCFG and LPDSCFG bits +SLEEPDEEP bit +WFI or WFE	Any external interrupt	Enable or be in low-power mode	All clocks in 1.2 V power domain disabled	HSICLK and HSECLK oscillators disabled
Standby	1.2 V power domain disabled	PDDSCFG bit + SLEEPDEEP bit +WFI or WFE	RTC alarm event, external reset on NRST pin IWDT reset	1.2 V voltage regulator disabled	Disable all clocks in 1.2 V power domain	HSICLK and HSECLK oscillators disabled

Sleep mode

The characteristics of sleep mode are shown in the table below.

Table 24 Characteristics of Sleep Mode

Characteristics	Description
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, the system can be awakened by any interrupt; if the WFE instruction is executed to enter sleep mode, it can be awakened by either an event or an interrupt.
Sleep	The core stops working, all peripherals are still running, and the data in the core registers and memory before sleep are saved.
Wakeup delay	None
After wake-up	If the system is awakened by an interrupt, it will first enter the interrupt state, then exit the interrupt, and then execute the program after WFI instruction. If the system is awakened by an event, it will directly execute the program after WFE instruction.

Stop mode

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

Table 25 Characteristics of Stop Mode

Characteristics	Description
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, 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. To reduce power consumption, be sure to set the LPDSCFG bit to "1" before the system enters Stop mode.
Wake-up	If WFI instruction is executed to enter the sleep mode, the system can be awakened by any interrupt; if the WFE instruction is executed to enter sleep mode, it can be awakened by either an event or an interrupt.
Stop	The core and the peripheral will stop working, and the data in the core register and memory before stop will be saved.
Wakeup delay	Wake-up time of HSICLK oscillator + wake-up time of voltage regulator from low-power mode.
After wake-up	If the system is awakened by an interrupt, it will first enter the interrupt state, then exit the interrupt, and then execute the program after WFI instruction. If the system is awakened by an event, it will directly execute the program after WFE instruction.

Standby mode

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

Table 26 Characteristics of Standby Mode

Characteristics	Description
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, enter the standby mode immediately.
Wake-up	The system is awakened by RTC alarm, or external reset on NRST pin and IWDT reset.
Standby	The core and the peripheral will stop working, and the data in the core register and memory will be lost.
Wakeup delay	Chip reset time.
After wake-up	The program starts executing from the beginning.

7.4.3.2 Low-power state

Table 27 Low-Power State

Name		Sleep state	Wake-up	Stop state	Wake-up	Standby state	Wake-up
Core		-	-	-	-	D	-
Clock	HSE	X	-	-	-	-	-
	HSI8M	X	-	-	-	-	-
	HSI14M	O	-	-	-	-	-
	LSI	O	-	O	-	-	-
	PLL	X	-	-	-	-	-
Anaog IP	PMU	Y	Y	Y	Y	Y	-
	3.3 V LDO	Y	-	N	-	L	-
	1.2 V LDO	Y	-	N	-	-	-
	1.6 V LDO	Y	-	-	-	-	-
	ADC	O	-	-	-	-	-
	TempSensor	O	-	-	-	-	-
	IO	Y	Y	Y	Y	Only NRST works	Only NRST can be awakened
Memory	Pflash	Y	-	Y	-	D	-
	DFLASH	Y	-	Y	-	D	-
	SRAM	Y	-	Y	-	D	-
3.3 V digital IP	RTC	O	Y	O	Y	O	Y
	IWDG	O	Y	O	Y	O	Y
Base IP	DMA	O	-	-	-	D	-
	GPIO	O	Y	-	Y	D	-
	EINT	O	Y	O	Y	D	-
	RCC	Y	Y	-	-	D	-
	SYSC	O	-	-	-	D	-
Timer	TMR1	O	Y	-	-	D	-
	TMR2	O	Y	-	-	D	-
	TMR3	O	Y	-	-	D	-
	TMR4	O	Y	-	-	D	-

Name		Sleep state	Wake-up	Stop state	Wake-up	Standby state	Wake-up
	TMR6	O	Y	-	-	D	-
	TMR7	O	Y	-	-	D	-
	TMR8	O	Y	-	-	D	-
Communication IP	SPI	O	Y	-	-	D	-
	USART1	O	Y	-	Y	D	-
	USART2	O	Y	-	Y	D	-
	CAN	O	Y	-	Y	D	-
Security IP	CRC	O	-	-	-	D	-
	TRNG	O	-	-	-	D	-
	AES256	O	-	-	-	D	-
	SHA256	O	-	-	-	D	-

Note: X: Indicates that at least one sleep mode is enabled; Y: normal operation; O: configurable (controlled by software)."-": not supported in this mode; N: Indicates the module is active or can enter a low-power mode; D: power-down; L: low-power.

7.4.3.3 Reduce the power consumption in run mode

In run mode, reduce power consumption by slowing down the system clock and disabling or reducing the APB/AHB peripheral clocks.

7.5 Register Address Mapping

Table 28 PWU Register Address Mapping

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

7.6 Register Functional Description

7.6.1 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 Deep Sleep Configuration Configure the working state of the voltage regulator in stop mode. 0: Enable 1: Low-power mode

Field	Name	R/W	Description
1	PDDSCFG	R/W	Power Down Deep Sleep Configure When the CPU enters deep sleep, configure the voltage regulator state in standby or stop mode. 0: The voltage regulator is controlled by LPDSCFG bit when entering the stop mode 1: Enter standby mode
2	WUFLGCLR	RC_W1	Clear Wake-up Flag 0: Invalid 1: Clear the wake-up flag by writing "1" after two system clock cycles
3	SBFLGCLR	RC_W1	Standby Flag Clear 0: Invalid 1: Write "1" to clear the standby flag
4	PVDEN	R/W	Enable Power Voltage Detector 0: Disable 1: Enable
7:5	PLSEL	R/W	Select PVD Voltage Threshold 0x0：2.73V 0x1：2.73V 0x2：3.05V 0x3：3.37V 0x4：3.66V 0x5：3.95V 0x6：4.26V 0x7：4.52V Note: See "Datasheet" for detailed instructions.
8	BPWEN	R/W	Enable RTC Domain Write Access The RTC domain includes RTC and its associated registers. Write access is prohibited after a system reset; writing 1 to this bit enables write access. 0: Disable write 1: Enable write
9	STOP_FILTER_EN	R/W	STOP_FILTER_EN: 1: Hardware-Controlled Filter Switching: The IO filter function is enabled. After the system enters Stop mode, the filtered IO input is used for wake-up. After the system wakes up from Stop mode, the IO automatically switches back to the non-filtered function. 0: Software-Controlled Filter Function: Enabling the IO filter function configures the IO as a filtered input; disabling the IO filter function configures the IO as a non-filtered input.
31:10	Reserved		

7.6.2 Power Control/Status Register (PMU_CSTS)

Offset address: 0x04

Reset value: 0x0000 000X (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 to indicate whether it is an RTC alarm wake-up event. 0: Not occurred 1: Occurred
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 CSF bit of the Power Control Register (PMU_CTRL). 0: Non-standby mode 1: Standby mode
2	PVDOFLG	R	PVD Output Flag It indicates whether V_{DD} /V_{DDA} 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} is higher than PVD threshold 1: V_{DD} /V_{DDA} is lower than PVD threshold Note: This bit is "0" after reset or when entering the standby mode (PVD stops work).
31:3	Reserved		

8 Nested Vector Interrupt Controller (NVIC)

8.1 Full Name and Abbreviation Description of Terms

Table 29 Full Names and Abbreviations of Terms

Full Name	Abbreviation
Non Maskable Interrupt	NMI

8.2 Introduction

The Arm® Cortex®-M0+ 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. See Arm® Cortex®-M0+ Technical Reference Manual for more instructions about NVIC.

8.3 Main Characteristics

(1) 32 maskable interrupt channels (excluding 16 Arm® Cortex®-M0+ interrupt lines)

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

(3) Power management control

(4) Low-delay exception and interrupt processing

(5) Implementation of system control register

8.4 Interrupt and Exception Vectors

Table 30 Interrupt and Exception Vectors

Name	Vector No.	Priority	Vector address	Description
-	-	-	0x0000_0000	Reserved
RST	-	-3	0x0000_0004	Reset
NMI	-	-2	0x0000_0008	Non-maskable interrupt
Hard fault	-	-1	0x0000_000C	Various hardware faults
SVCall	-	Can be set	0x0000_002C	System service called by general SWI instruction
PendSV	-	Can be set	0x0000_0038	Pending system service
SysTick	-	Can be set	0x0000_003C	System tick timer
-	-	-	0x0000_0040	Reserved

Name	Vector No.	Priority	Vector address	Description
PVD	1	Can be set	0x0000_0044	PVD comparator interrupt
RTC	2	Can be set	0x0000_0048	RTC interrupt
FLASH	3	Can be set	0x0000_004C	FLASH interrupt
RCM	4	Can be set	0x0000_0050	RCM interrupt
EINT0_1	5	Can be set	0x0000_0054	EINT line [1:0] interrupt
EINT2_3	6	Can be set	0x0000_0058	EINT line [3:2] interrupt
EINT4_15	7	Can be set	0x0000_005C	EINT line [15:4] interrupt
SMS	8	Can be set	0x0000_0060	SRAM ECC interrupt
DMA_CH1	9	Can be set	0x0000_0064	DMA channel 1 interrupt
DMA_CH2_3	10	Can be set	0x0000_0068	DMA Channel 2/3 interrupt
DMA_CH4_5	11	Can be set	0x0000_006C	DMA Channel 4/5 interrupt
ADC	12	Can be set	0x0000_0070	ADC interrupt
TMR1_BRK_UP_TRG_COM	13	Can be set	0x0000_0074	TMR1, BRK, UP, TRG and COM interrupt
TMR1_CC	14	Can be set	0x0000_0078	TMR1 capture/compare interrupt
TMR2	15	Can be set	0x0000_007C	TMR2 interrupt
TMR3	16	Can be set	0x0000_0080	TMR3 interrupt
TMR6	17	Can be set	0x0000_0084	TMR6 interrupt
TMR7	18	Can be set	0x0000_0088	TMR7 interrupt
TMR4	19	Can be set	0x0000_008C	TMR4 interrupt
-	20	Can be set	0x0000_0090	Reserved
SHA256	21	Can be set	0x0000_0094	SHA256 interrupt
AES256	22	Can be set	0x0000_0098	AES256 interrupt
CAN_IT0	23	Can be set	0x0000_009C	CAN interrupt line 0
CAN_IT1	24	Can be set	0x0000_00A0	CAN interrupt line 1
SPI	25	Can be set	0x0000_00A4	SPI interrupt
-	26	Can be set	0x0000_00A8	Reserved
USART1	27	Can be set	0x0000_00AC	USART1 interrupt
USART2	28	Can be set	0x0000_00B0	USART2 interrupt
TRNG	29	Can be set	0x0000_00B4	TRNG interrupt
TMR8	30	Can be set	0x0000_00B8	TMR8 interrupt
CAN_SMS_IT	31	Can be set	0x0000_00BC	CAN SRAM ECC interrupt and stop waken-up interrupt

9 External Interrupt and Event Controller (EINT)

9.1 Introduction

The interrupts/events are divided into internal interrupts/events and external interrupts/events. 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 implement 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 can output pulse by events such as GPIO, while the internal events trigger another TMR to work, for example, by an update event of a TMR.

9.2 Main Characteristics

(1) Support 32 event/interrupt requests

(2) Can be configured independently as the line of external/internal event request

(3) Each event/interrupt line can be masked independently

(4) The internal line is automatically disabled when the system is not in the stop mode

(5) Each external event/interrupt line can be triggered independently

(6) Each external interrupt line has dedicated status bit

(7) Simulate all external event interrupts

9.3 Functional Description

9.3.1 Classification and difference of "external interrupt and event"

"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 differences are shown in the table below:

Table 31 Classification and Differences of "External Interrupts and Events"

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 pending 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 event line, an event request pulse will be generated, and the corresponding pending bit will not be set to 1.
External software event	Software interrupt register/transmit 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 interrupt 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.

9.3.2 Core wake-up

Using WFI and WFE instructions can stop the core. When WFI instruction is used, any interrupt can wake up the core; when WFE instruction is used, the core can be woken up by an 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) Trigger an internal interrupt (internal hardware event) but do not trigger the interrupt handler function for wake-up.

- Enable an internal interrupt in the peripheral, but do not 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 by 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; after 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 pending bit NVIC interrupt channel.

9.3.3 External interrupt and event line mapping

Table 32 External Interrupt and Event Line Mapping

External Interrupt and Event Channel Name	External Interrupt and Event Line No.
PA0/PB0/PC0/PF0	EINT 0
PA1/PB1/PC1/PF1	EINT 1
PA2/PB2/PC2/PD2	EINT 2
PA3/PB3/PC3	EINT 3
PA4/PB4/PC4/PF4	EINT 4
…	…
PA7/PB7/PC7/PF7	EINT 7
PA8/PB8/PC8	EINT 8
…	…
PA15/PB15/PC15	EINT 15
PVD output	EINT 16
RTC alarm event	EINT 17
Reserved	EINT 18
Reserved	EINT 19
Reserved	EINT 20
Reserved	EINT 21
Reserved	EINT 22
Reserved	EINT 23
Reserved	EINT 24
Internal USART1 wake-up event	EINT 25
Reserved	EINT 26
Internal USART2 wake-up event	EINT 27
Reserved	EINT 28
Reserved	EINT 29
Reserved	EINT 30
Reserved	EINT 31

9.4 Register Address Mapping

Table 33 External Interrupt/Event Controller Register Address Mapping

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

9.5 Register Functional Description

9.5.1 Interrupt mask register (EINT_IMASK)

Offset address: 0x00

Reset value: 0x0F94 0000

Field	Name	R/W	Description
27:0	IMASKx	R/W	Mask Interrupt Request on Line x (x=0…27) 0: Mask 1: Open
31:28	Reserved		

9.5.2 Event mask register (EINT_EMASK)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
27:0	EMASKx	R/W	Mask Event Request on Line x (x=0…27) 0: Mask 1: Open
31:28	Reserved		

9.5.3 Enable the rising edge to trigger the register (EINT_RTEN)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
16:0	RTENx	R/W	Enable Rising Edge Trigger Event and Interrupt on Line x (x=0…16) 0: Disable 1: Enable
31:17	Reserved		

Note: Since the external wake-up lines are edge-triggered, there should be no glitch 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 pending bit will not be set; on the same interrupt line, the rising edge trigger and falling edge trigger can be set at the same time.

9.5.4 Enable the falling edge to trigger the register (EINT_FTEN)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
16:0	FTENx	R/W	Enable Falling Edge Trigger Event and Interrupt on Line x (x=0…16) 0: Disable 1: Enable
31:17	Reserved		

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

9.5.5 Software interrupt event register (EINT_SWINTE)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
16:0	SWINTEx	R/W	Software Interrupt Event on Line x (x=0…16) Set 1 by software, write 1 or clear 0 for 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)
31:17	Reserved		

9.5.6 Interrupt pending register (EINT_IPEND)

Offset address: 0x14

Reset value: 0xXXXX XXXX

Field	Name	R/W	Description
16:0	IPENDx	RC_W1	Flag of Interrupt Pending on Line x (x=0…16) Whether the selectable trigger request occurs 0: No 1: Occurred When a request is triggered by the corresponding edge of EINT_RTEN/EINT_FTEN on the external interrupt line, this bit is set to "1" by hardware; and cleared by writing "1" to this bit.
31:17	Reserved		

10 Direct Memory Access (DMA)

10.1 Full Name and Abbreviation Description of Terms

Table 34 Full Names and Abbreviations of Terms

Full Name	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 achieve direct 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 a DMA controller, with 5 channels in total. Each channel can manage multiple DMA requests, but each channel can only start 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) DMA has 5 channels

(2) There are three data transmission modes: peripheral to memory, memory to peripheral, and 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 for transmission is programmable, up to 65535

10.4 Functional Description

10.4.1 DMA request

If the peripheral or memory needs to transmit data using DMA, it is required to first transmit DMA request and after it is approved by DMA, data transmission can be started.

DMA has five channels. 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 trigger.

When multiple peripherals request the same channel, it is required to configure the corresponding register to enable or disable the request of each peripheral, so as to ensure that one channel can only enable one peripheral request.

Table 35 DMA Request Mapping

Channel	TMR1	TMR2	TMR3	TMR4	TMR6	TMR7	ADC	SPI	USART
Channel 1	—	TMR2_CH1	—	TMR4_CH3(1) TMR4_UP(1)	—	—	ADC(1)	—	—
Channel 2	TMR1_CH1	TMR2_CH4 TMR2_TRIG	TMR3_CH3	TMR4_CH2	—	—	ADC(2)	SPI1_RX	USART1_TX(1)
Channel 3	TMR1_CH2	TMR2_CH2	TMR3_CH4 TMR3_UP	TMR4_CH1	TMR6_UP	—	—	SPI1_TX	USART1_RX(1)
Channel 4	TMR1_CH4 TMR1_TRIG TMR1_COM	-	TMR3_CH1 TMR3_TRIG	TMR4_CH4 TMR4_TRIG	—	TMR7_UP	—	—	USART1_TX(2) USART2_TX
Channel 5	TMR1_CH3 TMR1_UP	TMR2_CH3 TMR2_UP	TMR3_CH2	TMR4_CH3(2) TMR4_UP(2)	—	—	—	—	USART1_RX(2) USART2_RX

Note:

(1) This DMA request is mapped to the DMA channel only when the corresponding bit of SYSCFG_CFG1 register is cleared.

(2) This DMA request is mapped to the DMA channel only when the corresponding remapping bit of SYSCFG_CFG1 register is set to "1".

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 MSIZE bit of DMA_CHCFGx register.

10.4.2.2 Transmission width and alignment method are programmable

DMA transmission operation of programmable data transmission width:

Figure 9 Transmission Width with Source of 8 Bits and Target of 8 Bits

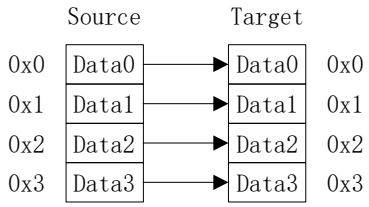


Figure 10 Transmission Width with Source of 8 Bits and Target of 16 Bits

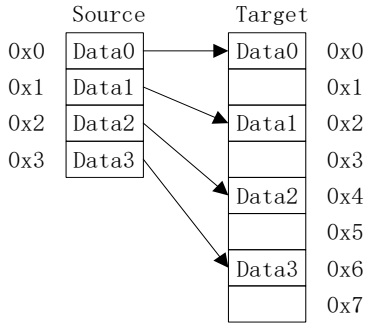


Figure 11 Transmission Width with Source of 8 Bits and Target of 32 Bits

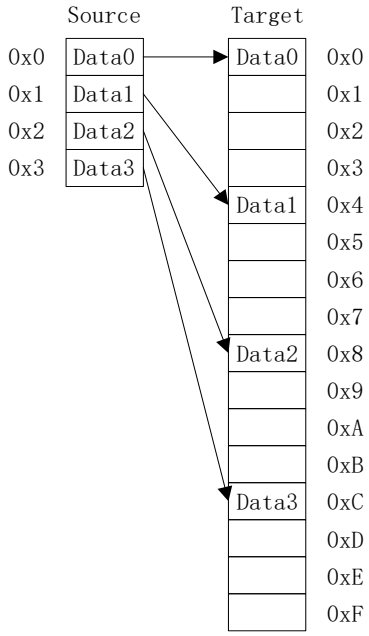


Figure 12 Transmission Width with Source of 32 Bits and Target of 8 Bits

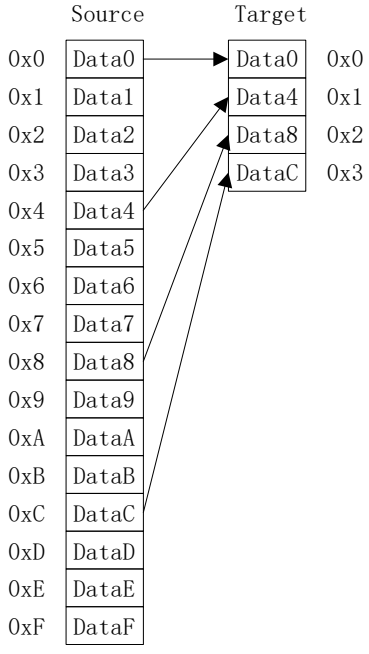


Figure 13 Transmission Width with Source of 16 Bits and Target of 16 Bits

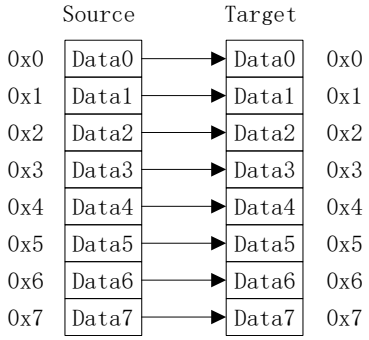


Figure 14 Transmission Width with Source of 16 Bits and Target of 32 Bits

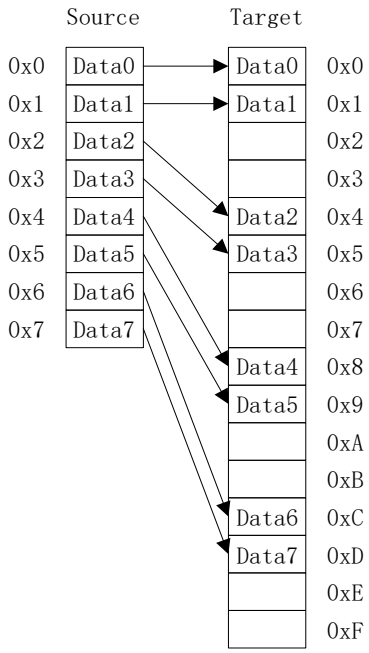
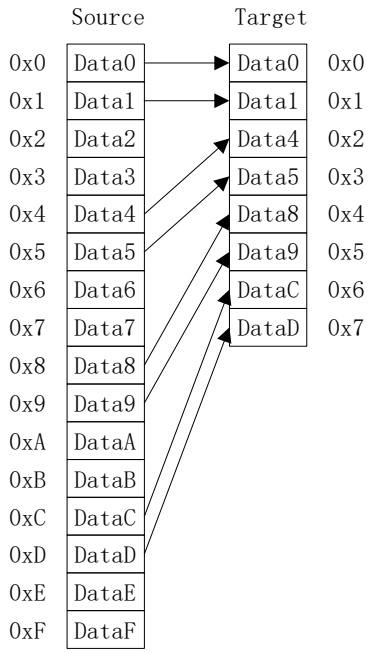


Figure 15 Transmission Width with Source of 32 Bits and Target of 16 Bits



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 by 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 ends, the DMA operation will not be performed any more, and a 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 the data transmission ends, 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 for transmission becomes 0, it will automatically return to the initial value and continue the DMA operation and when the CIRMODE bit is cleared, it will exit the circular mode.

10.4.2.5 DMA request priority setting

Arbiter

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 four priorities; 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 external RAM supported by internal SRAM (such as external SRAM); if the read operation (source address) is performed on the memory, the address includes internal FLASH and internal SRAM.

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

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

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 36 DMA Register Address Mapping

Register name	Description	Offset address
DMA_INTSTS	DMA interrupt status register	0x00
DMA_INTFCLR	DMA interrupt flag clear register	0x04
DMA_CHCFGx	DMA Channel x configuration register	0x08+20 x (channel number–1)
DMA_CHNDATAx	DMA Channel x transmission quantity register	0x0C+20 x (channel number–1)
DMA_CHPADDRx	DMA Channel x peripheral address register	0x10+20 x (channel number–1)
DMA_CHMADDRx	DMA Channel x memory address register	0x14+20 x (channel number–1)

10.6 Register Functional Description

10.6.1 DMA interrupt status register (DMA_INTSTS)

Offset address: 0x00

Reset value: 0x0000 0000

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

Field	Name	R/W	Description
19, 15,11,7,3	TERRFLGx	R	Channel x Transfer Error Occur Flag (x=1...5) Indicate whether the transmission error interrupt (TE) is generated on the channel; these bits are set to 1 by hardware; write 1 and clear 0 on the corresponding bit of DMA_INTFCLR. 0: Not generate 1: Generate
31:20	Reserved		

10.6.2 DMA interrupt flag clear register (DMA_INTFCLR)

Offset address: 0x04

Reset value: 0x0000 0000

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

10.6.3 Configuration register of DMA Channel x (DMA_CHCFGx) (x=1…5)

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

Field	Name	R/W	Description
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 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
11:10	MSIZE	R/W	Memory Data Size Configure 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved
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 Transmission quantity register of DMA Channel x (DMA_CHNDATAx) (x=1…5)

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 become read-only, indicating the number of remaining bytes to be transmitted. The register will decrease every time 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 enabled or not.
31:16	Reserved		

10.6.5 DMA Channel x peripheral address register (DMA_CHPADDRx) (x=1…5)

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

Reset value: 0x0000 0000

This register cannot be written when the channel is enabled (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…5)

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

Reset value: 0x0000 0000

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

Field	Name	R/W	Description
31:0	MEMADDR	R/W	Memory Basic Address Setup When MSIZE= '01' (16 bits) and MEMADDR[0] bit is not used, it will be aligned with 16-bit address automatically during transmission. When MSIZE= '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 37 Full Names and Abbreviations of DBGMCU Terms

Full Name	Abbreviation
Frame Clock	FCLK
Data Watchpoint Trigger	DWT
Break Point Unit	BPU

11.2 Introduction

G32A1085 A1065 A1045 series product uses Arm® Cortex® -M0+ core which 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.

Supported debugging interface: Serial interface.

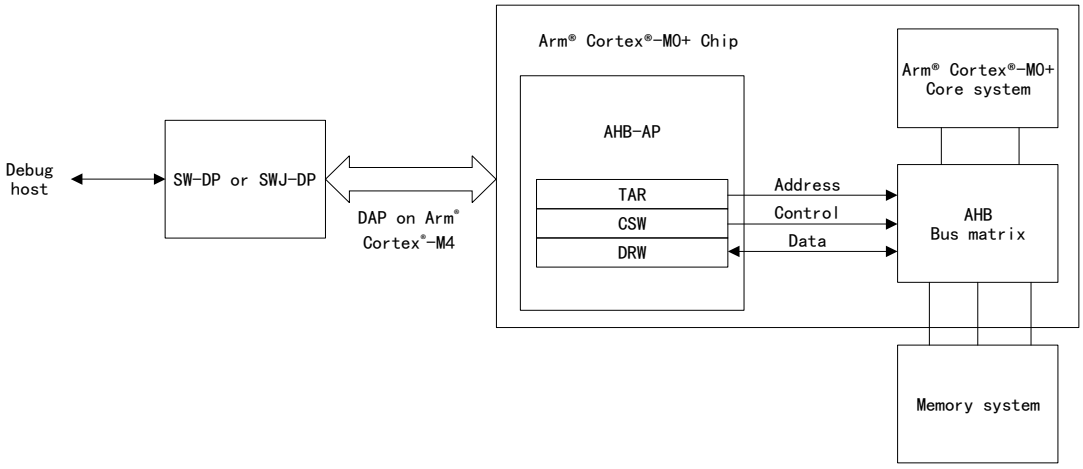
Note: The hardware debug interface included in Arm® Cortex® -M0+ core is a subset of Arm CoreSight development tool set. Refer to Arm® Cortex® -M0+ (Version r1p1) Technical Reference Manual (TRM) and CoreSight Development Tool Set (Version r1p0) TRM for more information about debug function of Arm® Cortex® -M0+ core.

11.3 Main Characteristics

(1) Flexible debug pin assignment

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

Figure 16 G32A1085 A1065 A1045 Level and Arm ® Cortex ® -M0+ Level Debugging Block Diagram



11.4 Functional Description

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

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

(3) Flexible implementation of production of offline programmer

11.5 Register Address Mapping

Table 38 DBGMCU Register Address Mapping

Register name	Description	Address
DBGMCU_IDCODE	Debug MCU device ID register	0x4001 5800
DBGMCU_CFG	Debug MCU configuration register	0x4001 5804
DBGMCU_APB1F	Debug MCU APB1 freeze register	0x4001 5808
DBGMCU_APB2F	Debug MCU APB2 freeze register	0x4001 580C

11.6 Register Functional Description

11.6.1 Debug MCU device ID register (DBGMCU_IDCODE)

Address: 0x4001 5800

Reset value: 0xXXXX XXXX

Only support 32-bit access

Field	Name	R/W	Description
11:0	EQR	R	Equipment Recognition This field indicates device ID. G32A1085 series: 0x015 G32A1065 series: 0x022 G32A1045 series: 0x023 The debugger/programming tool identifies chips by EQR (11:0).
15:12	Reserved (reading return 0xA)		
31:16	WVR	R	Wafer Version Recognition This field indicates device version. G32A1085 A1065 A1045 series product: Version A: 0x0024

11.6.2 Debug MCU configuration register (DBGMCU_CFG)

Address: 0x4001 5804

Reset value: 0x0000 0000 (unaffected by system reset)

It only supports 32-bit access.

This register allows configuring MCU during debugging and supports low-power mode.

It is reset asynchronously by POR (not reset by system), and can be written by debugger through system reset.

If the debugging master does not support these characteristics, the user software can write to these registers.

Field	Name	R/W	Description
0	Reserved		
1	STOP_CLK_STS	R/W	Configure Debug Stop Mode 0: In the stop mode when both FCLK and HCLK are disabled, all clocks will be disabled by clock controller. When exiting the stop mode, the clock configuration is the same as that after reset (the clock is provided by the 8 MHz internal RC oscillator HSICLK), so the software needs to reconfigure the clock controller to enable PLL, crystal oscillator, etc. 1: In the stop mode when both FCLK and HCLK are enabled, both FCLK and HCLK are provided by internal RC oscillator. The internal RC oscillator remains or is active in the stop mode. When it exits the stop mode, the software must reconfigure the clock controller to enable PLL, crystal oscillator, etc.

Field	Name	R/W	Description
2	STANDBY_CLK_STS	R/W	Debug Standby Mode 0: When both FCLK and HCLK are disabled, the digital part is not powered on. From the software level, it indicates that when the MCU just exits the standby mode, others exit the debug mode, which is the same as reset. 1: When both FCLK and HCLK are enabled, the digital part is powered on, and the internal RC oscillator provides FCLK and HCLK clocks. Besides, the MCU exits the standby mode by system reset, which is the same as reset.
31:3	Reserved		

11.6.3 Debug MCU APB1 freeze register (DBGMCU_APB1F)

Address: 0x4001 5808

Reset value: 0x0000 0000 (unaffected by system reset)

It only supports 32-bit access.

This register is used to configure MCU during debugging.

Involved APB peripherals:

- Freeze the timer counter
- Freeze supporting system window regulators and independent watchdog counter

This register is reset asynchronously by POR (not reset by system) and can be written by the debugger through system reset.

Field	Name	R/W	Description
0	TMR2_STS	R/W	Configure TMR2 Work Status When Core Stops Whether TMR2 counter continues to work when the core stops work. 0: Continue to work 1: Stop working
1	TMR3_STS	R/W	Configure Timer3 Work Status When Core Stops Whether TMR3 counter continues to work when the core stops work. 0: Continue to work 1: Stop working
3:2	Reserved		
4	TMR6_STS	R/W	Configure Timer6 Work Status When Core Stops Whether TMR6 counter continues to work when the core stops work. 0: Continue to work 1: Stop working
7:5	Reserved		
8	TMR4_STS	R/W	Configure Timer4 Work Status When Core Stops Whether TMR4 counter continues to work when the core stops work. 0: Continue to work 1: Stop working

Field	Name	R/W	Description
9	Reserved		
10	RTC_STS	R/W	Configure RTC Work Status When Core Stops Whether RTC counter continues to work when the core stops work. 0: Continue to work 1: Stop working
11	Reserved		
12	IWDT_STS	R/W	Configure Independent Watchdog Work Status When Core Stops. Whether IWDT continues to work when the core stops 0: Continue to work 1: Stop working
31:13	Reserved		

11.6.4 Debug MCU APB2 freeze register (DBGMCU_APB2F)

Address: 0x4001 580C

Reset value: 0x0000 0000(unaffected by system reset)

It only supports 32-bit access.

This register is used to configure MCU during debugging.

Involved APB peripheral:

- Freeze the timer counter

This register is reset asynchronously by POR (not reset by system) and can be written by the debugger through system reset.

Field	Name	R/W	Description
10:0	Reserved		
11	TMR1_STS	R/W	Configure TMR1 Work Status When Core Stops Whether TMR1 counter continues to work when the core stops. 0: Continue to work 1: Stop working
15:12	Reserved		
16	TMR7_STS	R/W	Configure TMR7 Work Status When Core Stops Whether TMR7 counter continues to work when the core stops. 0: Continue to work 1: Stop working
17	TMR8_STS	R/W	Configure TMR8 Work Status When Core Stops Whether TMR8 counter continues to work when the core stops work. 0: Continue to work 1: Stop working
31:18	Reserved		

12 General-Purpose/Alternate Function I/O Pin (GPIO/AFIO)

12.1 Full Name and Abbreviation Description of Terms

Table 39 Full Names and Abbreviations of Terms

Full Name	Abbreviation
P-channel Metal Oxide Semiconductor	P-MOS
N-channel Metal Oxide Semiconductor	N-MOS

12.2 Main Characteristics

The GPIO ports can be configured for the following functions through 32-bit configuration registers (GPIOx_MODE / GPIOx_OMODE / GPIOx_OSSEL / GPIOx_PUPD) and two 32-bit data registers (GPIOx_IDATA / GPIOx_ODATA):

(1) Input mode
- 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

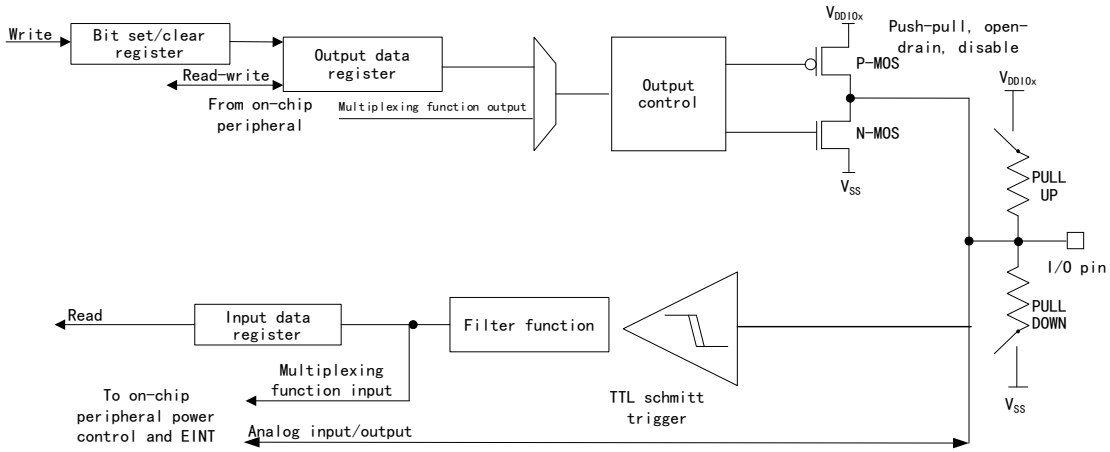
(4) Analog mode

(5) GPIO can be used as external interrupt/wakeup line

(6) Support locking I/O configuration function

12.3 Structure Block Diagram

Figure 17 GPIO Structure Block Diagram



Note: Theoretically, whether the input signal is filtered does not affect the GPIO functions.

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 and input mode and multiplexing function by software. All GPIO interfaces have external interrupt capability.

12.4.1 IO status during reset and just after reset

During and just after GPIO reset, if the multiplexing function is not enabled, the I/O port will be configured as floating input mode.

After reset, the debug pin is in AF pull-up/pull-down state:

- PA14: SWCLK in pull-up mode
- PA13: SWDIO in pull-down mode

12.4.2 Input mode

In input mode, the GPIO can be configured as pull-up, pull-down, floating, or analog input. The filter function can be optionally enabled or disabled. Theoretically, whether the input signal is filtered does not affect the fundamental operation of the GPIO.

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

Pull-up, pull-down, and floating modes

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

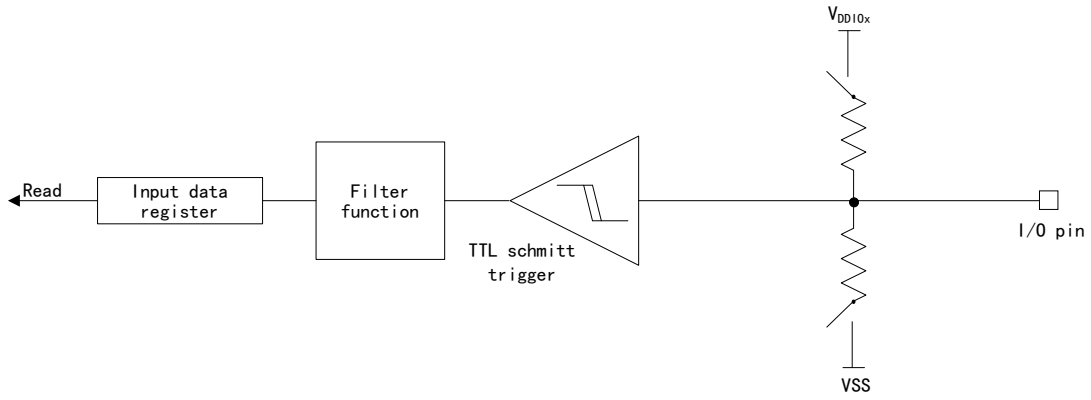
- Schmitt trigger is enabled

- Disable output buffer
- By configuring the pull-up/pull-down register GPIOx_PUPD, select whether to use pull-up/pull-down resistor
- The input data register GPIOx_IDATA captures the data on I/O pin in each AHB clock cycle.
- Read I/O state by 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 if pull-up, and low if pull-down; when connecting the equipment, it is determined by the external input level and load impedance.

Figure 18 I/O Structure in Input Mode



12.4.3 Output mode

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

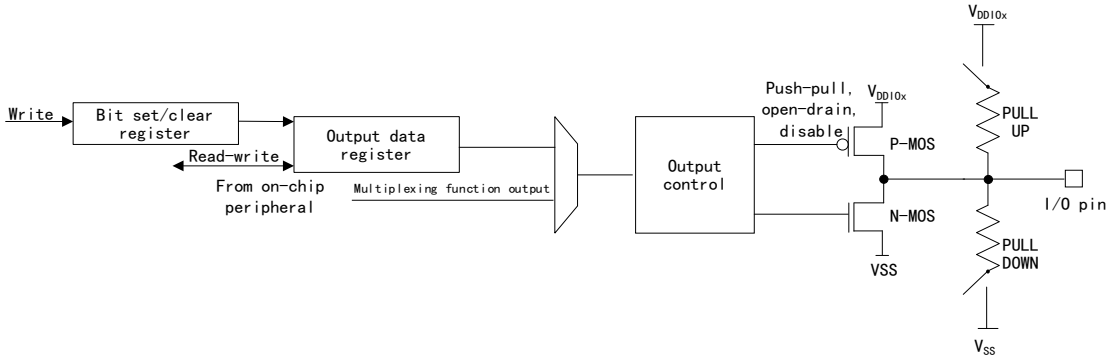
When GPIO is configured as the output pin, the output speed of the port can be configured and the output drive mode (push-pull/open-drain) can be selected.

In output mode：

- Schmitt trigger is enabled
- Activate output buffer
- By configuring the pull-up/pull-down register GPIOx_PUPD, select whether to use pull-up/pull-down resistor
- 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

- The input data register GPIOx_IDATA captures the data on I/O pin in each AHB clock cycle
- Read the actual I/O state through the input data register GPIOx_IDATA

Figure 19 I/O Structure in Output Mode



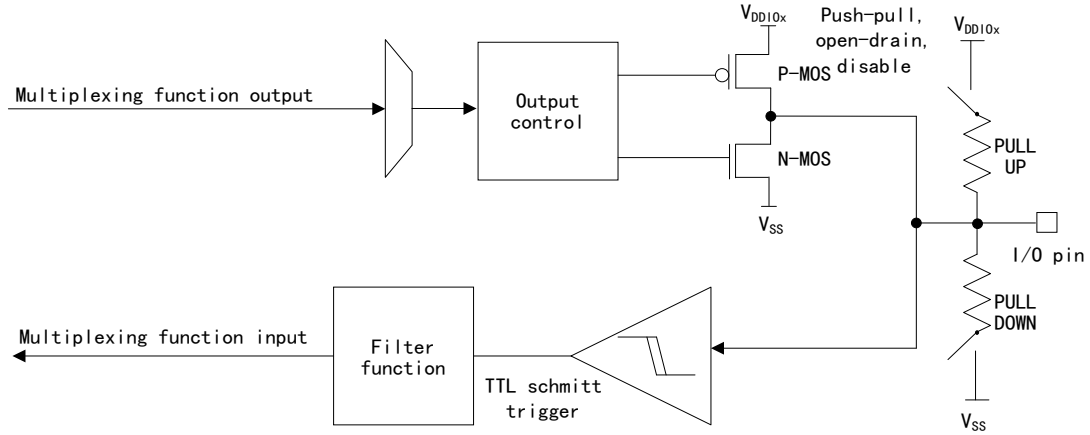
12.4.4 Multiplexing mode

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

In push-pull/open-drain multiplexing mode:

- Enable the output buffer
- Output buffer is driven by peripheral
- Activate Schmitt trigger input
- By configuring the pull-up/pull-down register GPIOx_PUPD, select whether to use pull-up/pull-down resistor
- The data on the I/O pin is sampled in each AHB clock cycle and stored in the port input status register
- Read the actual I/O state through the input data register GPIOx_IDATA

Figure 20 I/O Structure in Multiplexing Mode

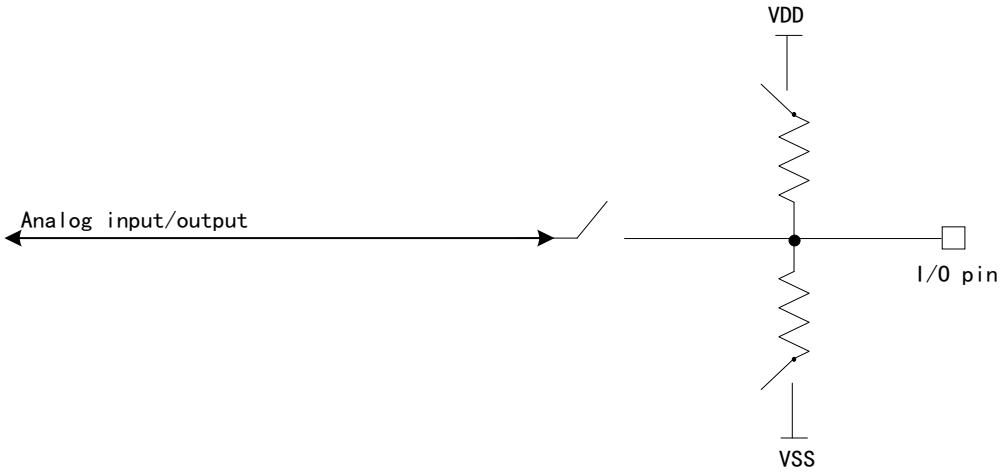


12.4.5 Analog mode

In analog function 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
- Read the value of the input data register to be 0

Figure 21 Analog Function I/O Port Structure



12.4.6 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.

12.4.7 I/O data bit processing

GPIO port set/reset register (GPIOx_BSC) allows set/reset operation for each bit of the output data register (GPIOx_ODATA). The valid data width of the set/reset register is double the valid data width of GPIOx_ODATA.

Writing 0 to any bit in GPIOx_BSC will not affect the value of the GPIOx_ODATA register. BSy and BCy bits of GPIOx_BSC are set to 1 for the same time, operation of BSy bit has the priority. The GPIOx_BSC register can modify the corresponding bits in the GPIOx_ODATA register. Setting a bit in the BS (Bit Set) field of GPIOx_BSC sets the corresponding bit in GPIOx_ODATA to 1, while setting a bit in the BC (Bit Clear) field of GPIOx_BSC clears the corresponding bit in GPIOx_ODATA.

When using the set or reset access mechanism of GPIOx_BSC register, it is not necessary to disable interrupts via software when accessing GPIOx_ODATA.

12.4.8 Multiplexing function and remapping

Multiplexer

The multiplexer is used to connect the I/O port line of the device to the embedded peripheral module, and it can only be one-to-one at the same time.

Each I/O pin is equipped with a multiplexer. The multiplexer has up to 16 multiplexing inputs, but in fact it uses up to 7 (AF0-AF6) inputs, which are configured by GPIOx_ALFL and GPIOx_ALFH registers. When I/O pin is reset, all pin ports are connected to AF0.

Remapping

Each peripheral has multiple multiplexing functions, but only one multiplexing function input can be selected for a pin, so the multiplexing function of the peripheral can be mapped to other I/O pins, that is, the multiplexing function signal can be reassigned to a pin address.

The pin multiplexing function and the remapping address table can be found in the pin multiplexing list in the Datasheet.

I/O multiplexing configuration

When I/O port is connected to the peripheral multiplexing function, the following debugging needs to be done:

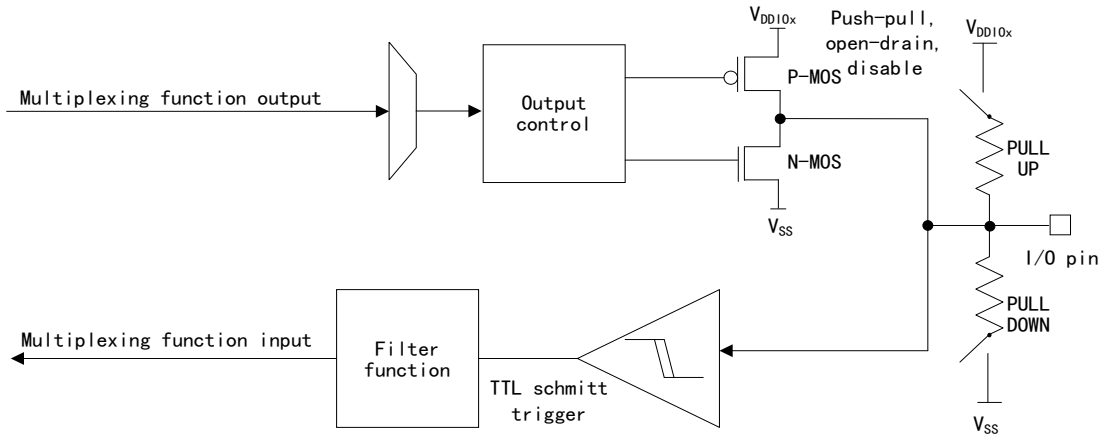
- After reset, the pin is configured with multiplexing function
- I/O port is configured as input, output or analog input
- The I/O port is connected to the defined AFx
- Configure pin pull-up/pull-down and output speed
- Configure I/O as multiplexing function in GPIOx_MODE

When the I/O port is configured with multiplexing function, its input and output mode is as follows:

- Enable the output buffer
- Output buffer is driven by peripheral
- Activate Schmitt trigger input
- By configuring the pull-up/pull-down register GPIOx_PUPD, select whether to use pull-up/pull-down resistor
- The data on the I/O pin is sampled in each AHB clock cycle and stored in the port input status register
- Read the actual I/O state through the input data register GPIOx_IDATA

The multiplexing mode I/O structure is shown in the figure below:

Figure 22 I/O Structure in Multiplexing Mode



12.4.9 GPIO locking function

The locking mechanism of GPIO can protect the configuration of I/O port.

Write sequence (specific) to GPIOx_LOCK register so as to freeze the control register of Port A and Port B. If you want to write GPIOx_LOCK register, a specific write/read sequence should be transmitted.

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.

Each GPIOx_LOCK bit freezes the corresponding bits in the control registers (GPIOx_MODE, GPIOx_OMODE, GPIOx_OSSEL, GPIOx_PUPD, GPIOx_AFRL, and GPIOx_AFRH).

The LOCK sequence (Lock Register (GPIOx_LOCK) (x = A…B) is configured for the GPIO Port) must be performed exclusively via word (32-bit) access to the GPIOx_LOCK register. This is because Bit 16 and [15:0] bits of GPIOx_LOCK must be set simultaneously.

12.4.10 HSECLK pin served as GPIO

Enable or disable HSECLK RC oscillator via configuring HSEEN in RCM_CTRL1 and RCM_BDCTRL registers.

When HSECLK RC oscillator is enabled, the oscillator controls the related pins, and the related pins are unrelated to GPIO configuration; when HSECLK RC oscillator is disabled, the related oscillators can be used as general GPIO port.

12.5 Register Address Mapping

Table 40 GPIO Register Address Mapping

Register name	Description	Offset address
GPIOx_MODE	Port mode register	0x00
GPIOx_OMODE	Port output mode register	0x04
GPIOx_OSSEL	Port output drive capability register	0x08
GPIOx_PUPD	Port pull-up/pull-down register	0x0C
GPIOx_IDATA	Port bit input data register	0x10
GPIOx_ODATA	Port bit output data register	0x14
GPIOx_BSC	Port set/reset register	0x18
GPIOx_LOCK	Port lock register	0x1C
GPIOx_ALFL	Port multiplexing function low-8-bit register	0x20
GPIOx_ALFH	Port multiplexing high-8-bit register	0x24
GPIOx_BR	Port reset register	0x28
GPIOx_FILTER_EN	Filter Enable Register	0x30

12.6 Register Functional Description

12.6.1 Port mode register (GPIOx_MODE) (x=A…D, F)

Offset address: 0x00

Reset value: 0x2800 0000 for Port A

0x0000 0000 for other ports

Field	Name	R/W	Description
31:0	MODEy[1:0]	R/W	Configure Port x Pin y Mode (y=0…15) 00: Input mode (state after reset) 01: General output mode 10: Multiplexing function mode 11: Analog mode

12.6.2 Port output mode register (GPIOx_OMODE) (x=A…D, F)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	OMODEy	R/W	Configure Port x Pin y Output Mode (y=0…15) 0: Push-pull output (reset state) 1: Open-drain output
31:16	Reserved		

12.6.3 Port output drive capabiilty register (GPIOx_OSSEL) (x=A…D, F)

Offset address: 0x08

Reset value: 0x0C00 0000 for Port A

0x0000 0000 for other ports

Field	Name	R/W	Description
31:0	OSSELy[1:0]	R/W	Select Port x Pin y Output Speed (y=0…15) 00: Low speed, low drive capability 01: Middle speed, low drive capability 10: Low speed, high drive capability 11: High speed, high drive capability The speed and drive of I/O port is written by software

12.6.4 Port pull-up/pull-down register (GPIOx_PUPD) (x=A…D, F)

Offset address: 0x0C

Reset value: 0x2400 0000 for Port A

0x0000 0000 for other ports

Field	Name	R/W	Description
31:0	PUPDy[1:0]	R/W	Configure Port x Pin y Pull-up/Pull-down (y=0…15) These bits are written by software to configure pull-up/pull-down of the port bit 00: Pull-up/Pull-down is disabled 01: Pull up 10: Pull down 11: Reserved

12.6.5 Port input data register (GPIOx_IDATA) (x=A…D, F)

Offset address: 0x10

Reset value: 0x0000 XXXX

Field	Name	R/W	Description
15:0	IDATAy	R	Port x Pin y Input Data (y=0…15) These bits can only be read to store the input values of the corresponding I/O ports.
31:16	Reserved		

12.6.6 Port output data register (GPIOx_ODATA) (x=A…D, F)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	ODATAy	R/W	Port x Pin y Output Data (y=0…15) Read and write operations can be performed by software. For atomic bit setting/clearing, the ODATAy bit can be set separately by writing to GPIOx_BSC or GPIOx_BR register.
31:16	Reserved		

12.6.7 Port set/reset register (GPIOx_BSC) (x=A…D, F)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	BSy	W	Port x Pin y Set Bit (y=0…15) These bits can only be written, and the value of 0x0000 is returned when reading these bits. These bits are used to affect the corresponding ODATAy bits 0: No effect 1: Set the corresponding ODATAy bit
31:16	BCy	W	Port x Pin y Reset Bit (y=0…15) These bits can only be written, and the value of 0x0000 is returned when reading these bits. These bits are used to affect the corresponding ODATAy bits 0: No effect 1: Clear corresponding ODATAy bit If BSy bit and BCy bit are set at the same time, BSy has the priority

12.6.8 Port lock register (GPIOx_LOCK) (x=A\B)

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

Offset address: 0x1C

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	LOCKy	R/W	Configure Port x Pin y Lock Bit (y=0…15) 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.

Field	Name	R/W	Description
16	LOCKKEY	R/W	Lock Key This bit determines whether the port configuration lock key bit is activated 0: Not activated 1: Activate; GPIOx_LOCK register is locked until the next MCU reset is generated. Lock key write sequence: Write LOCK[16]=1+LOCK[15:0] Write LOCK[16]=0+LOCK[15:0] Write LOCK[16]=1+LOCK[15:0] Read LOCK Read LOCK[16]=1 (this read operation can be selected to confirm whether to activate the lock key) Note: The value of LOCKy cannot be changed in the write sequence of operation lock key. Any error in the write sequence of operation lock key will abort the lock. After the first lock sequence on any bit of the port, any read access on the LOCKKEY bit will return "1" until the next MCU is reset or the peripheral is reset.
31:17	Reserved		

12.6.9 Port multiplexing function low 8-bit register (GPIOx_ALFL) (x=A…D, F)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	ALFSELy	R/W	Select the alternate function of Port x Pin y (y=0…7) (Port x Pin y Multiplexing Function Select) These bits can be read by software to configure the multiplexing function of the port. ALFSELy selection: 0000：AF0 0001：AF1 0010：AF2 0011：AF3 0100：AF4 0101：AF5 0110：AF6 0111：AF7 1xxx: Reserved

12.6.10 Port multiplexing function high 8-bit register (GPIOx_ALFH) (x=A…F)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	ALFSELy	R/W	Port x Pin y Multiplexing Function Select (y=8…15) These bits can be read by software to configure the multiplexing function of the port. ALFSELy selection: 0000：AF0 0001：AF1 0010：AF2 0011：AF3 0100：AF4 0101：AF5 0110：AF6 0111：AF7 1xxx: Reserved

12.6.11 Port reset register (GPIOx_BR) (x=A…D, F)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	BRy	W	Configure Port x Pin y Reset (y=0…15) These bits can only be written, and the returned value is 0x0000 when reading these bits. These bits are used to affect the corresponding ODATAy 0: No effect 1: Clear corresponding ODATAy bit
31:16	Reserved		

12.6.12 Filter enable register (GPIOx_FILTER_EN)(x= A…D, F)

Offset address: 0x30

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	FILTER_ENy	R/W	Enable Port x Pin y Filter (y=0…15) 0: Disable IO filter 1: Enable IO filter
31:16	Reserved		

13 Timer Overview

13.1 Full Name and Abbreviation Description of Terms

Table 41 Full Names and Abbreviations of Terms

Full Name	Abbreviation
Timer	TMR
Update	U
Request	R
Event	EV
Capture	C
Compare	C
Length	LEN

13.2 Timer Category and Main Difference

This series of products contains three types of timers: advanced timer, general-purpose timer and basic timer (watchdog timer is 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 without external interface.

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

Table 42 Main Differences among Timers Included in the Product

Item	Specific content/Category	Advanced timer	General-purpose timer			Basic timer		
Name	—	TMR1	TMR2	TMR3	TMR4	TMR6	TMR7	TMR8
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	UP
Channel	Input channel	4		0	0
	Capture /Compare channel	7	4	0	0
	Complementary output channel	3	0	0	0
Function	Generate DMA request	Yes		Yes	No
	PWM mode	Yes		No	No
	Single-pulse mode	Yes		Yes	Yes
	Forced output mode	Yes		No	No
	Dead zone insertion	Yes	No	No	No

Timer Terms

Table 43 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 44 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 compare 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

14 Advanced Timer (TMR1)

14.1 Introduction

The advanced timer TMR1 takes the time base unit as the core, with the functions of input capture, output compare and braking input, and has a 16-bit autoload counter. Compared with other timers, the advanced timer supports complementary output, repeat count and programmable deadband insertion function, and is more suitable for motor control.

14.2 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
- Autoreload 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 deadband insertion

(5) Timing function

(6) Braking function

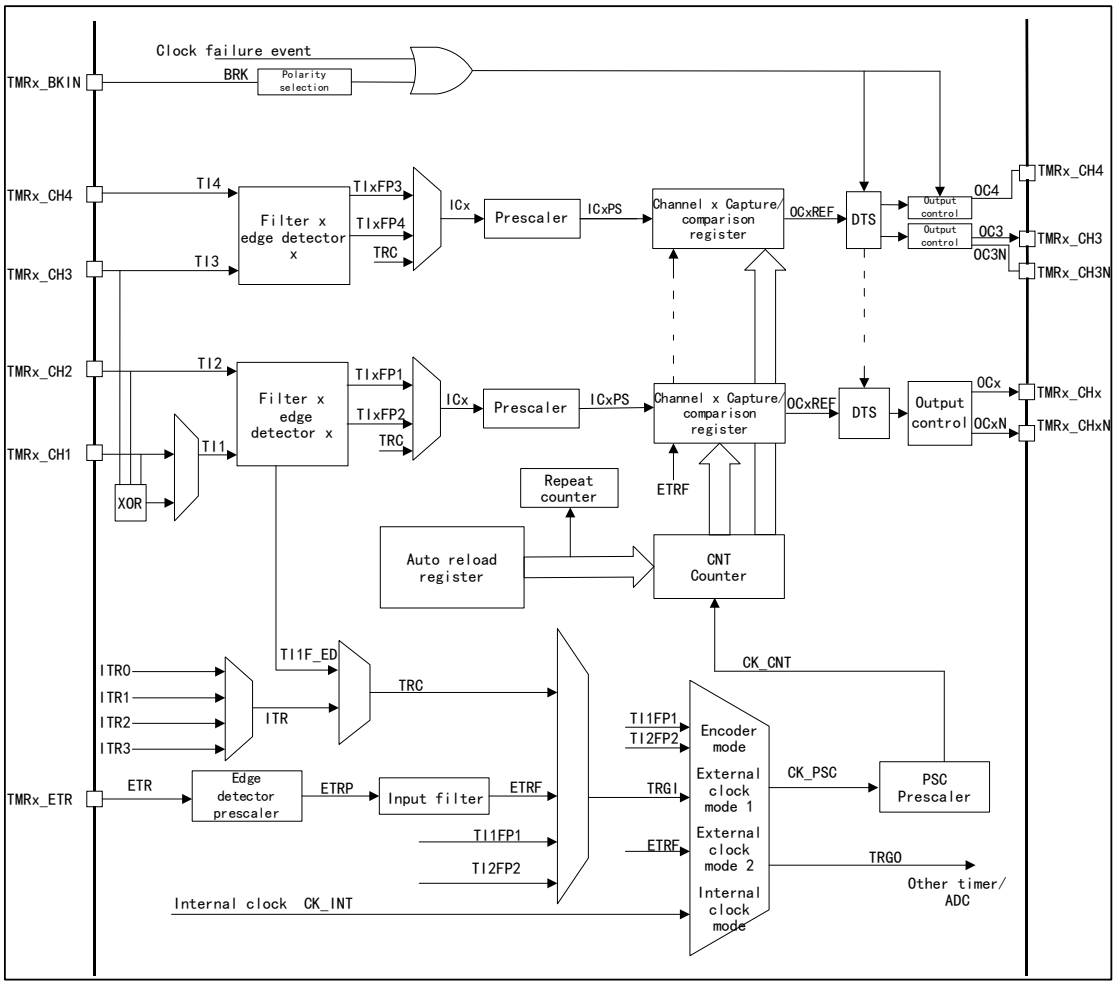
(7) Master/Slave mode controller of timer
- Timers can be synchronized and cascaded
- Support multiple slave modes and synchronization signals

(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

14.3 Structure Block Diagram

Figure 23 TMR1 Structure Block Diagram



14.4 Functional Description

14.4.1 Clock source selection

The advanced timer has four clock sources:

Internal clock

It is TMR1_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 input channels TI1/2/3/4 from the timer itself generate trigger signals after polarity selection and filtering, which are then connected to the slave mode controller to control the operation of the counter. In particular, the input of channel 1 is processed through both rising and falling edge detection, and the resulting pulse signals are logically OR to produce the TI1F_ED signal, also known as the TIF_ED dual-edge signal. PWM input can only be provided through the TI1 or TI2 inputs.

External clock mode 2

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

14.4.2 Timebase unit

The timebase unit in the advanced timer contains four registers:

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

Counter CNT

There are three count modes for the counter in the advanced timer:

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

Count-up mode

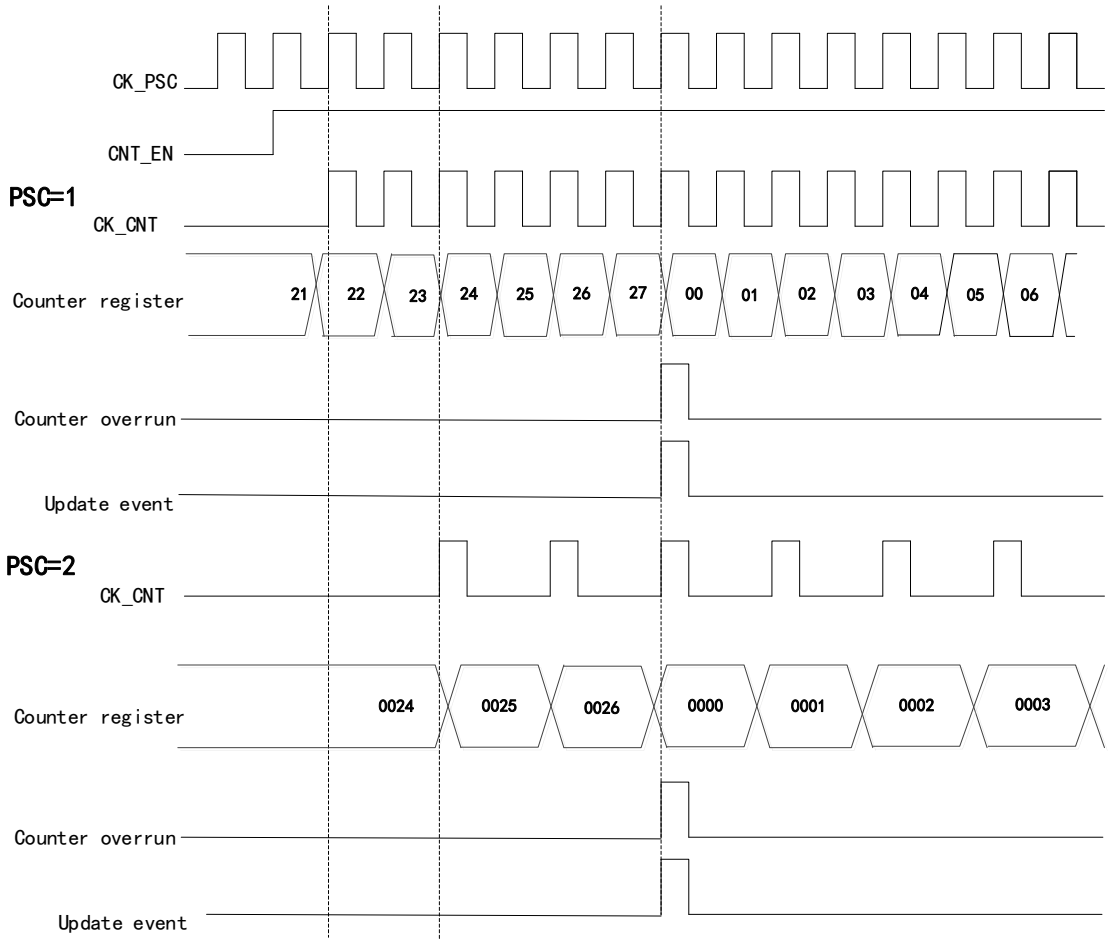
Set to the count-up mode by configuring CNTDIR bit of control register (TMR1_CTRL1).

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 (TMR1_CNT) is equal to the value of the auto reload (TMR1_AUTORLD), the counter will start to count from 0 again, a count-up overrun event will be generated, and the value of the auto reload (TMR1_AUTORLD) is written in advance.

When the counter overruns, an update event will be generated. At this time, the repeat count shadow register, the auto reload shadow register and the prescaler buffer will be updated. The update event can be disabled by configuring UD bit of control register TMR1_CTRL1.

The figure below is the timing diagram of count-up mode when the division factor is 1 or 2:

Figure 24 Timing Diagram of Count-up Mode when Division Factor is 1 or 2



Count-down mode

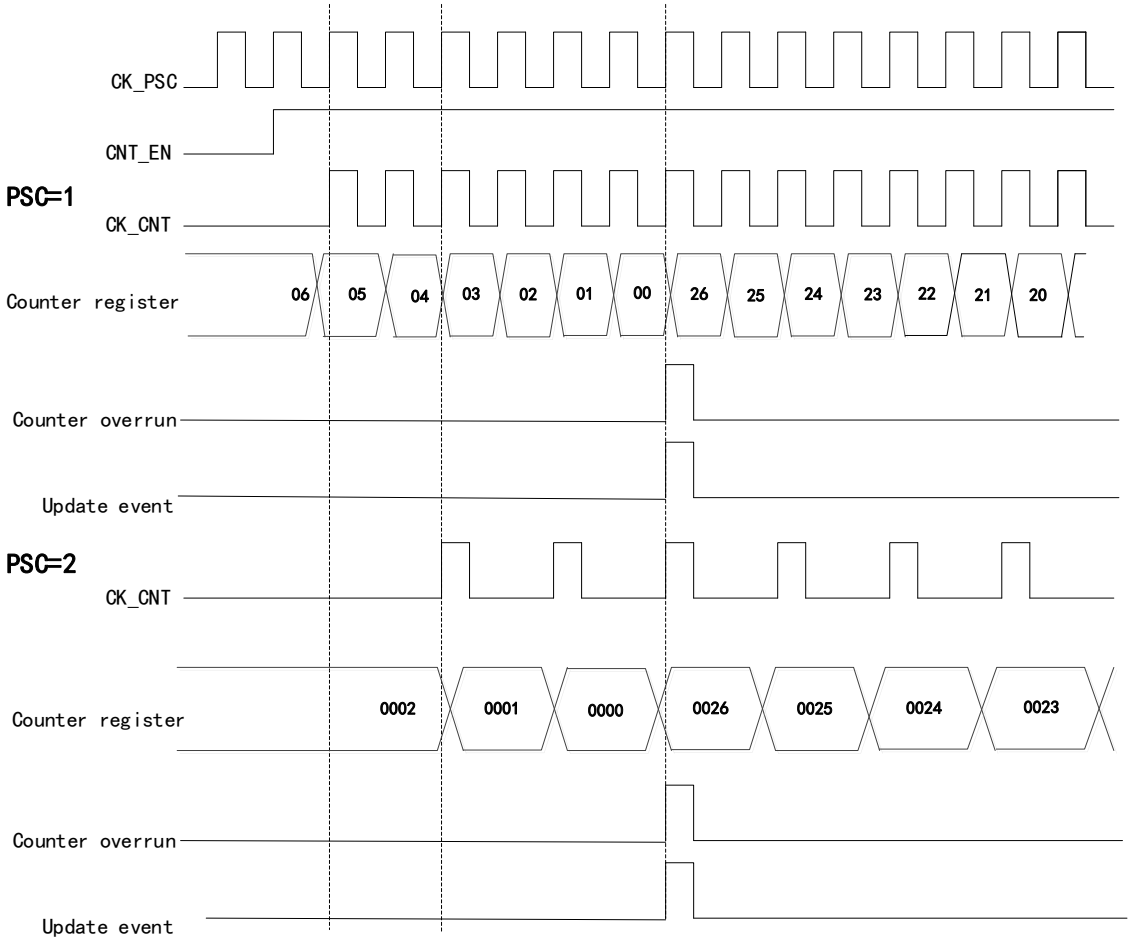
Set to the count-down mode by configuring CNTDIR bit of control register (TMR1_CTRL1).

When the counter is in count-down mode, it will start to count down from the value of the auto reload (TMR1_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 (TMR1_AUTORLD), meanwhile, a count-down overrun event will be generated, and the value of the auto reload (TMR1_AUTORLD) is written in advance.

When the counter overruns, an update event will be generated. At this time, the

repeat count shadow register, 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 TMR1_CTRL1 register.

Figure 25 Timing Diagram of Count-down Mode when Division Factor is 1 or 2

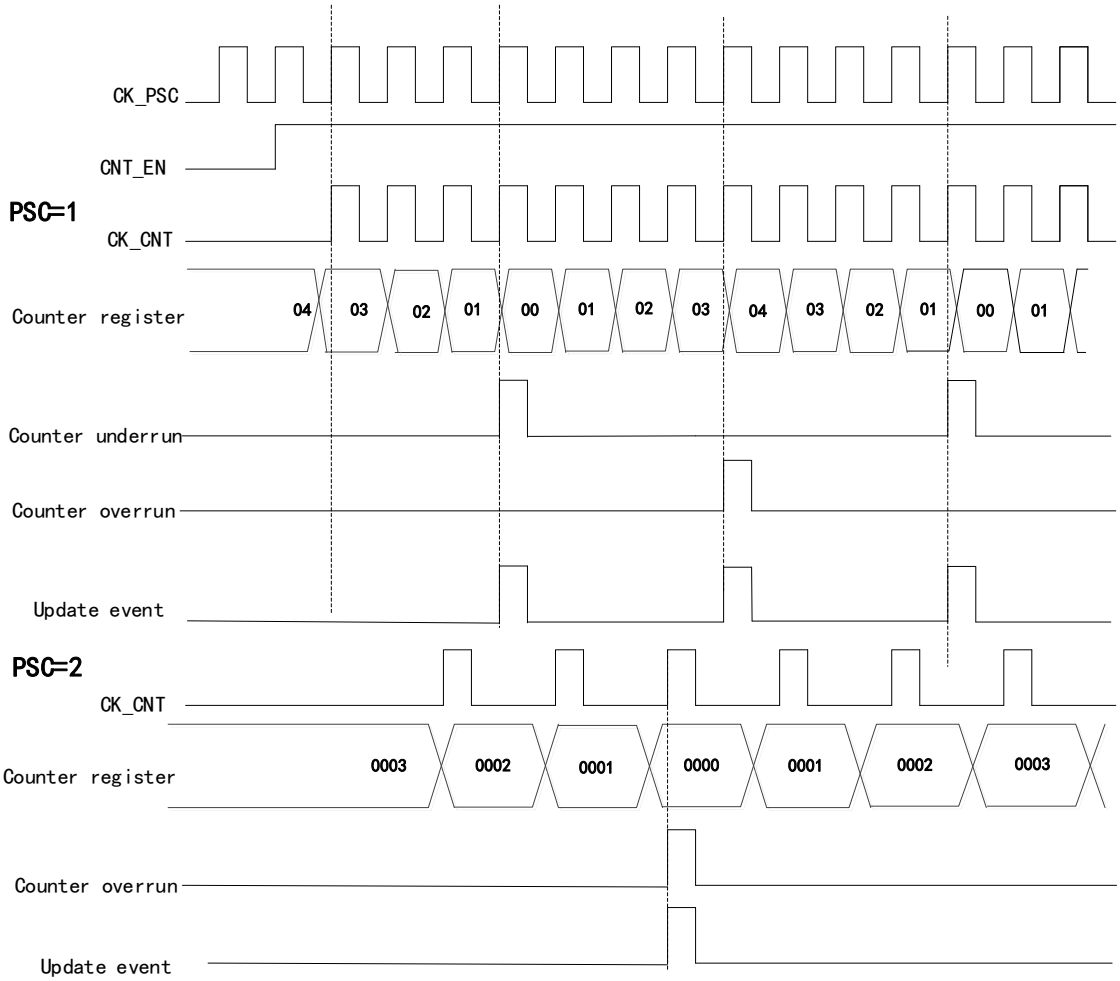


Center-aligned mode

Set to the center-aligned mode by configuring CNTDIR bit of control register (TMR1_CTRL1).

When the counter is in center-aligned mode, the counter counts up from 0 and when it reaches the value of auto reload (TMR1_AUTORLD), it counts down to 0 from the value of the auto reload (TMR1_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 26 Timing Diagram of Center-Aligned Mode when Division Factor is 1 or 2



Repeat counter REPCNT

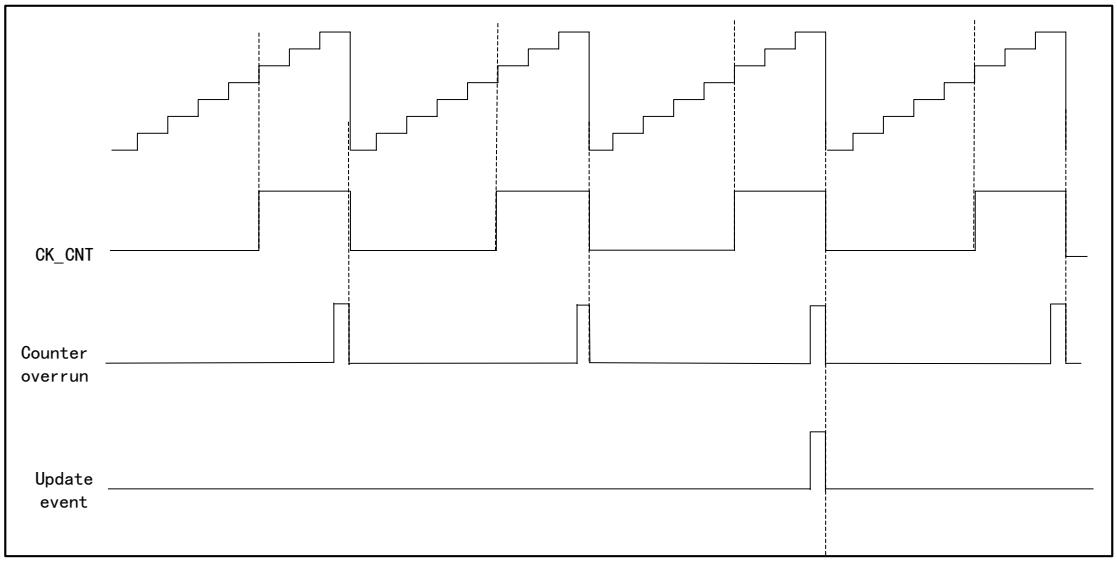
There is no repeat counter REPCNT in the basic/general-purpose timer, which means that when an overrun event or underrun event occurs in the basic/general-purpose timer, an update event will be generated directly; while in the advanced timer, because of the existence of the repeat counter, when an overrun/unerrrun 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 decrease by 1, and an update event will be generated when 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 27 Timing Diagram of Count-up Mode when Setting REPCNT=2



Prescaler PSC

The prescaler is 16 bits and programmable, and it can divide the clock frequency of the counter to any value within 1~65536 (controlled by TMR1_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.

14.4.3 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 enter the capture channels. 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 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 TMR1_CCx register will capture the current value of the counter and the CCxIFLG bit of the status register TMR1_STS will be set to 1; if CCxIEN=1, an interrupt will be generated.

In capture mode, the timing, frequency, cycle 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; read the value of capture register and the cycle of this pulse signal will be obtained by capture.

14.4.4 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, PWM mode 1 and PWM mode 2, which are configured by OCxMOD bit in TMR1_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 TMR1_CCMx register and the CCxPOL bit in the output polarity TMR1_CCEN register.

When CCxIFLG=1 in TMR1_STS register, if CCxIEN=1 in TMR1_DIEN register, an interrupt will be generated; if CCDSEL=1 in TMR1_CTRL2 register, a DMA request will be generated.

14.4.5 PWM output mode

PWM mode is pulse signal that can be adjusted by external output of 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 PWM Mode 1 when CCx=5, AUTORLD=7:

Figure 28 Timing Diagram of PWM1 Count-up Mode

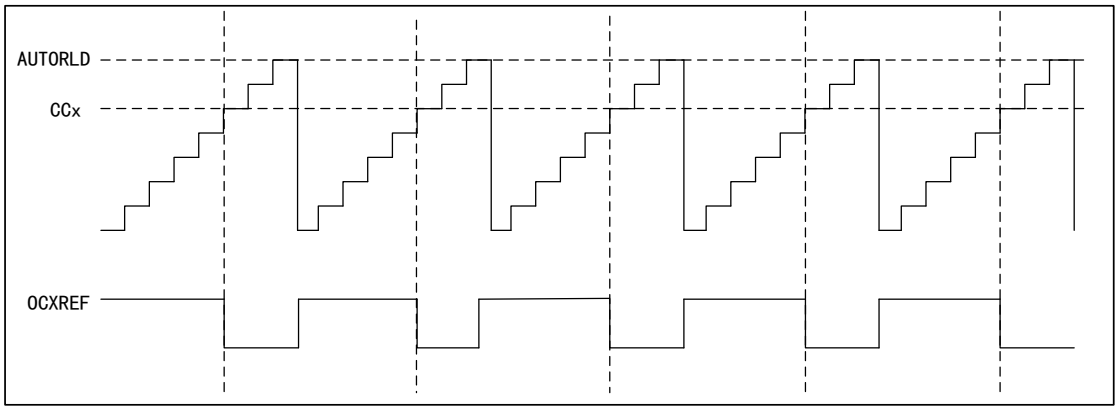


Figure 29 Timing Diagram of PWM1 Count-down Mode

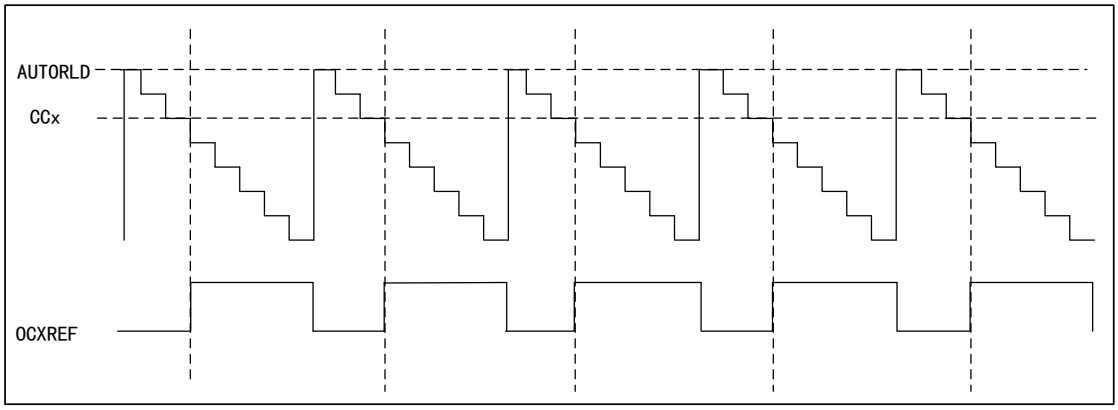
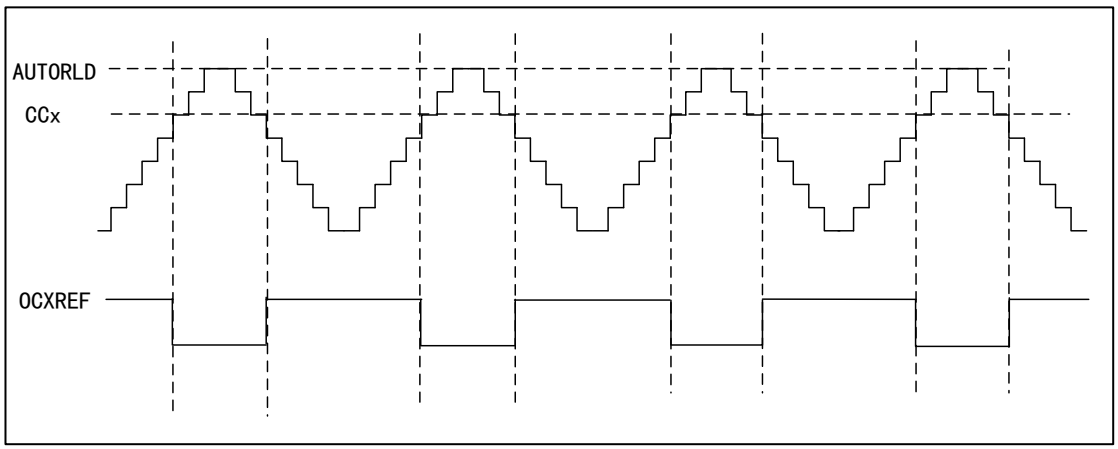


Figure 30 Timing Diagram of PWM1 Center-Aligned Mode



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 31 Timing Diagram of PWM2 Count-up Mode

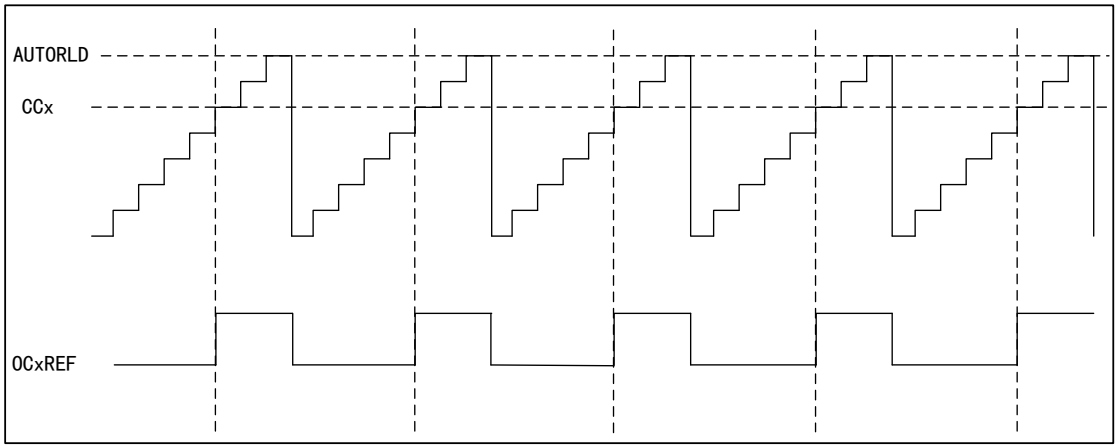


Figure 32 Timing Diagram of PWM2 Count-down Mode

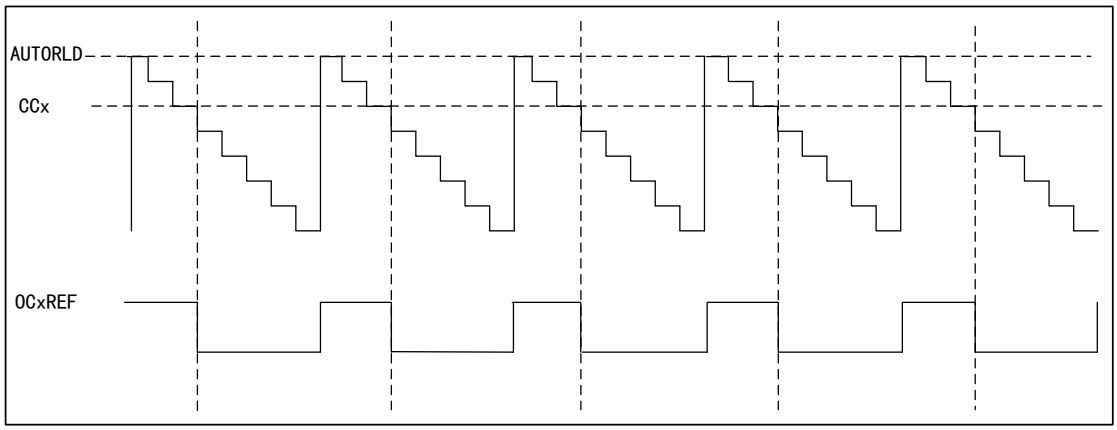
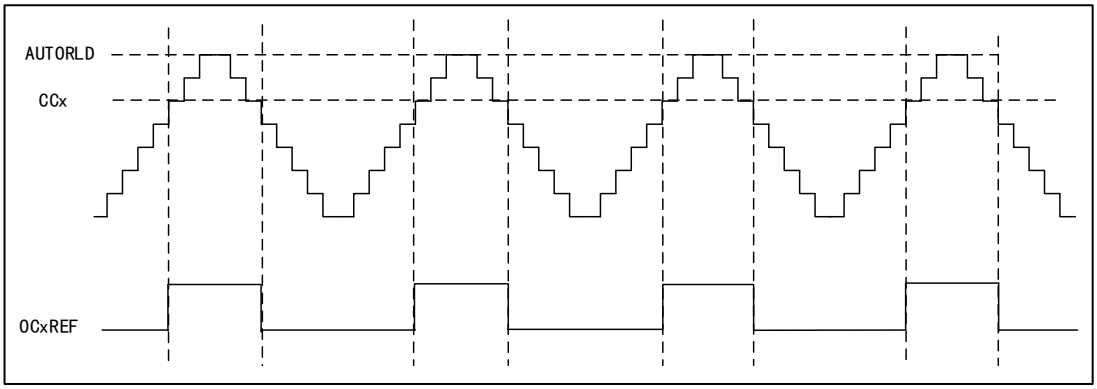


Figure 33 Timing Diagram of PWM2 Center-Aligned Mode



14.4.6 PWM input mode

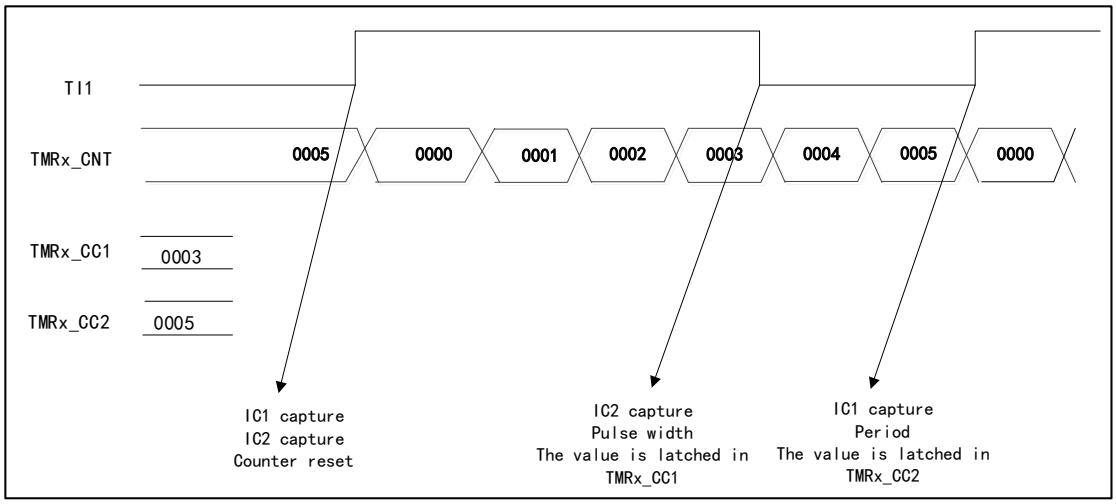
PWM input mode is a particular case of input capture.

In PWM input mode, as only TI1FP1 and TI1FP2 are connected to the slave mode controller, input can be performed only through the channels TMR1_CH1 and TMR1_CH2, which need to occupy the capture registers of CH1 and CH2.

In the PWM input mode, the PWM signal enters from TMR1_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 in TMR1_SMCTRL register)

Figure 34 Timing Diagram in PWM Input Mode



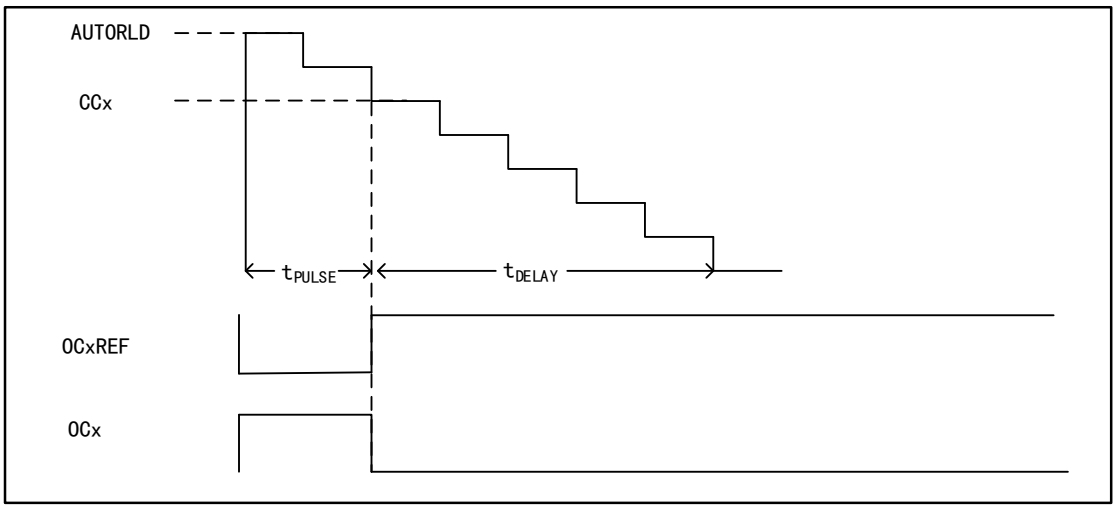
14.4.7 Single-pulse mode

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

Set SPMEN bit of TMR1_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 TMR1_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 35 Timing Diagram of Single-Pulse Mode



14.4.8 Impact of the register on output waveform

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

(1) CCxEN and CCxNEN bits in TMR1_CCEN register

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

(2) MOEN bit in TMR1_BDT register

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

(3) OCxOIS and OCxNOIS bits in TMR1_CTRL2 register

- OCxOIS=0 and OCxNOIS=0: When idle (MOEN=0), the output level after the deadband is 0
- OCxOIS=1 and OCxNOIS=1: When idle (MOEN=0), the output level after the deadband is 1

(4) RMOS bit in TMR1_BDT register

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

(5) IMOS bit in TMR1_BDT register

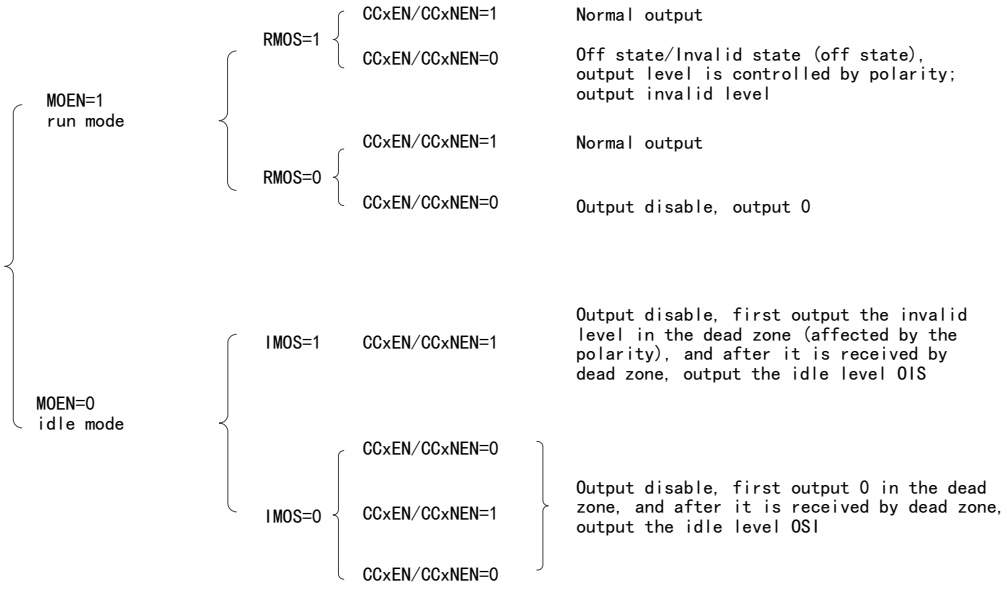
- Application environment of IMOS: In idle mode of corresponding complementary channel and timer (MOEN=0), the timer is not working (CCxEN=0, CCxNEN=0) or is working (CCxEN=1, CCxNEN=1)

(6) CCxPOL and CCxNPOL bits of TMR1_CCEN register

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

The following figure lists the register structural relationships that affect the output waveform.

Figure 36 Register Structural Relationship Affecting Output Waveform



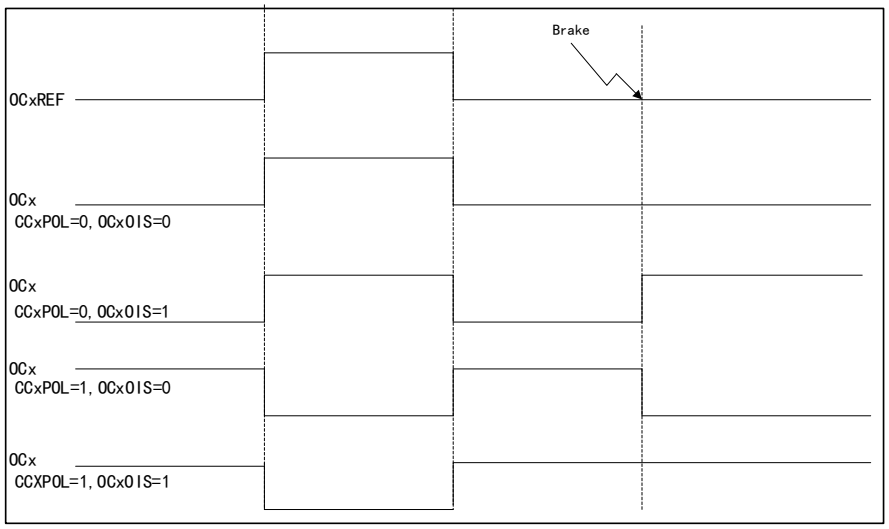
14.4.9 Braking function

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

Besides, the BRKEN bit in TMR1_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 37 Braking Event Timing Diagram

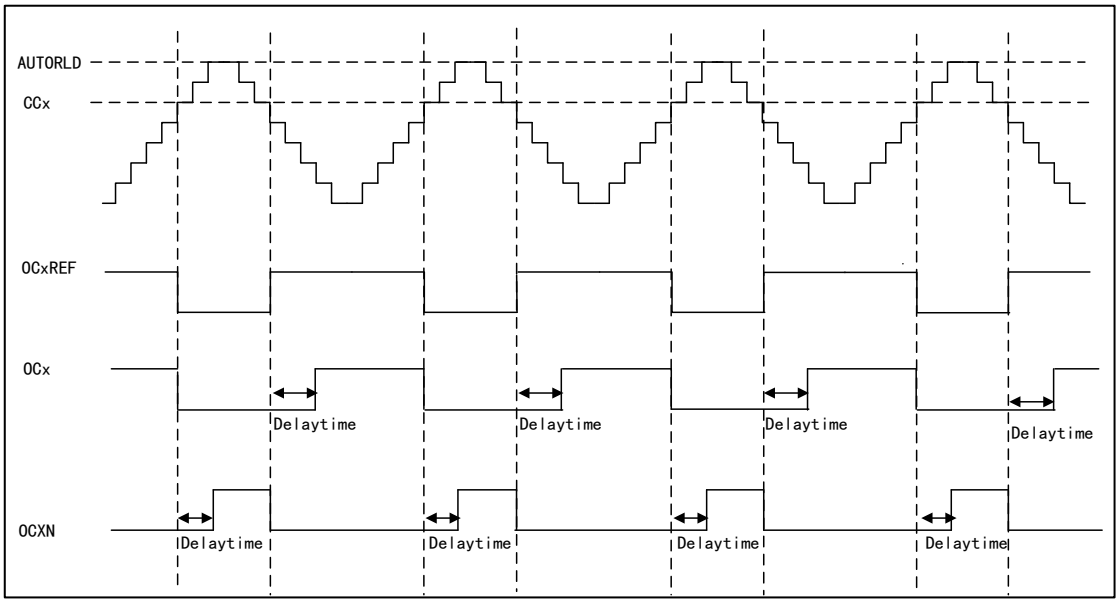


14.4.10 Complementary output and deadband insertion

TMR1 is configured with three pairs of complementary output channels. The insertion deadband 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 deadband is set according to the output device connected to the timer and its characteristics.

The duration of the deadband can be controlled by configuring DTS bit of TMR1_BDT register.

Figure 38 Complementary Output with Deadband Insertion



14.4.11 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 TMR1_CCMx register, set CCx channel as output
- OCxMOD=100/101 for TMR1_CCMx register, set to force OCxREF signal to invalid/valid

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

14.4.12 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 method of selecting encoder interface is as follows:

- Set the counter to count on the edge of TI1 channel or TI2 channel, or count on the edges of both TI1 and TI2 at the same time by setting SMFSEL bit of TMR1_SMCTRL register.
- Select the polarity of TI1 and TI2 by setting the CC1POL and CC2POL bits of TMR1_CCEN register.
- Select to filter or not by setting the IC1F and IC2F bits of TMR1_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 TMR1_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 45 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

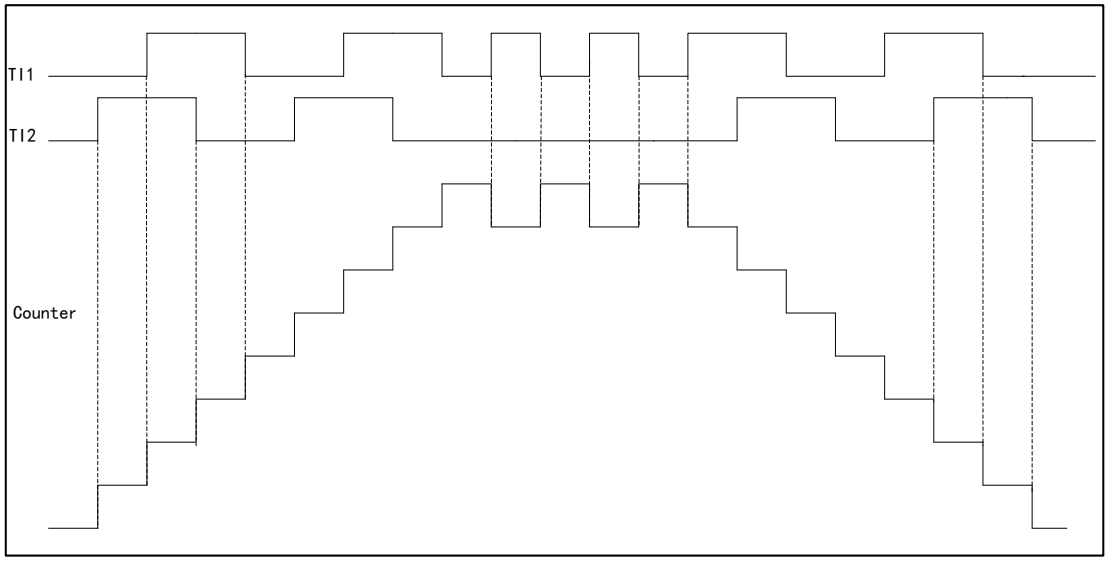
Effective edge		Count only in TI1		Count only in TI2	Count in both TI1 and TI2	
	Falling Edge	Count down	Count up		Count down	Count up

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 to noise interference.

Among the following examples:

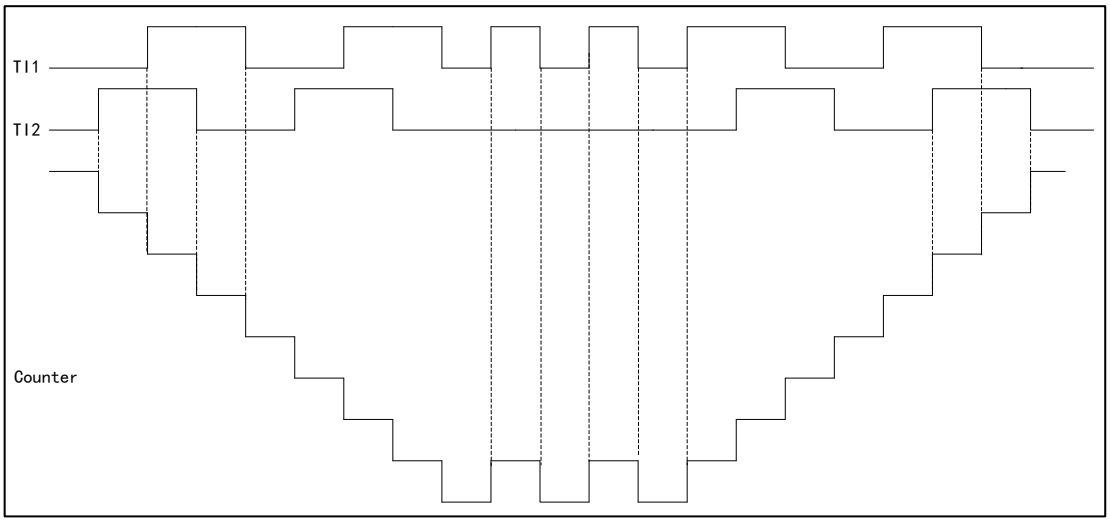
- IC1FP1 is mapped to TI1
- IC2FP2 is mapped to TI2
- Neither IC1FP1 nor IC2FP2 is phase-inverting
- The input signal is valid at the rising edge and falling edge
- Enable the counter

Figure 39 Operation Instance of Counter in Encoder Mode



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

Figure 40 Encoder Interface Mode Instance with IC1FP1 Inverted



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

14.4.13 Slave mode

TMR1 timer can synchronize external trigger:

- Reset mode
- Gated mode
- Trigger mode

SMFSEL bit in TMR1_SMCTRL register can be set to select the mode.

SMFSEL=100 set the reset mode, SMFSEL=101 set the gated mode, and 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 end. When the trigger input is high, the clock of the counter will be enabled. Once the trigger input becomes low, the counter will stop (but not be reset). The start and stop of the counter are controlled.

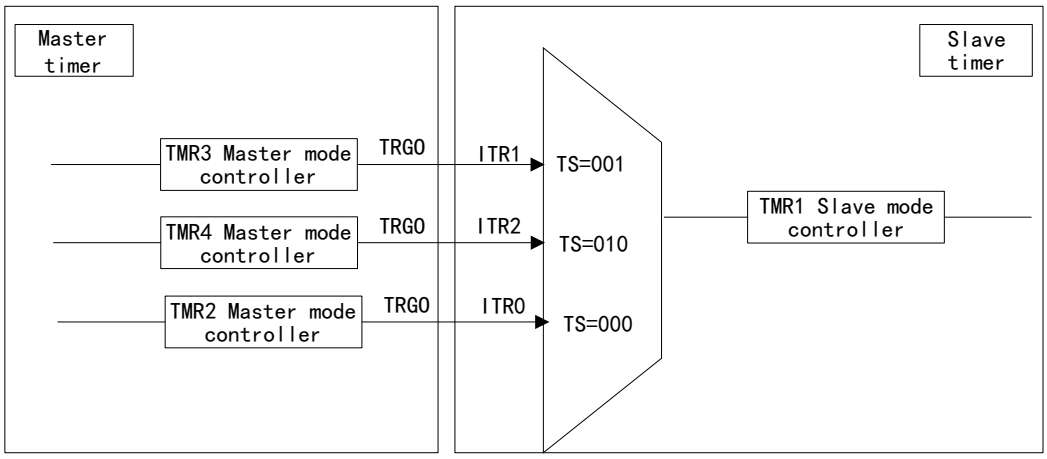
In the trigger mode, the enable of the counter depends on the event on the selected input, the counter will be enabled at the rising edge of the trigger input (but not be reset), and only the start of the counter is controlled.

14.4.14 Timer interconnection

Each timer of TMR1 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 41 Timer 1 Master/Slave Mode Instance



When the timers are interconnected:

- A timer can be used as the prescaler of other register
- Start the other register by the enable signal of a timer
- Start the other register by the update event of a timer
- Enable one timer to select another timer
- Two timers can be synchronized by an external trigger

14.4.15 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 trigger DMA requests.

14.4.16 Clear OCxREF signal when an external event occurs

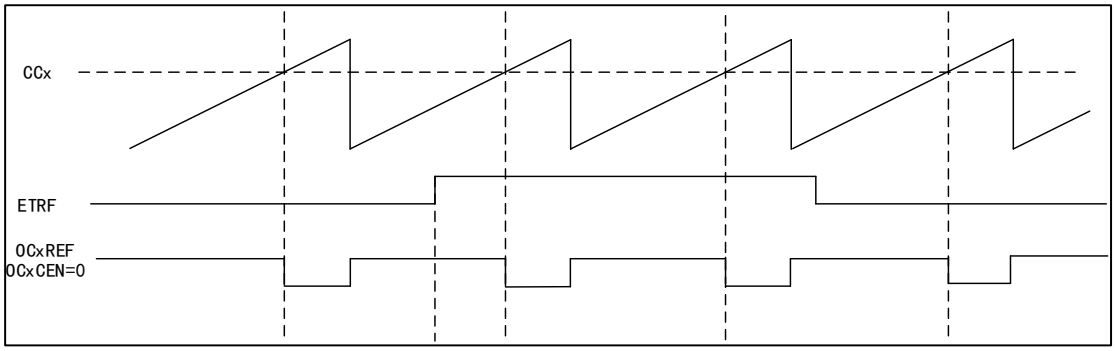
This function is used for output compare and PWM mode.

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 TMR1_CCMx is set to 1, and OCxREF signal will remain low level until the next update event occurs.

Set TMR1 to PWM mode, disable the external trigger prescaler, and disable the external trigger mode 2; when ETRF input is high, set OCxCEN=0, and the

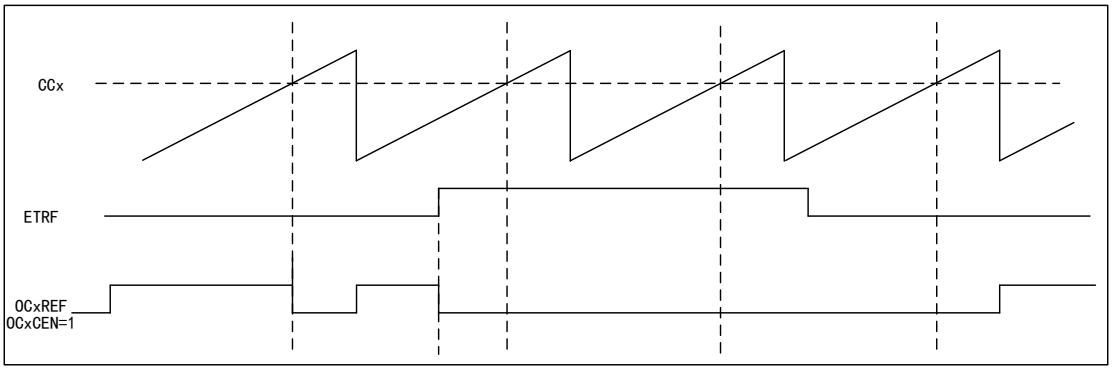
output OCxREF signal is shown in the figure below.

Figure 42 OCxREF Timing Diagram



Set TMR1 to PWM mode, disable 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 43 OCxREF Timing Diagram



14.5 Register Address Mapping

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

Table 46 TMR1 Register Address Mapping

Register name	Description	Offset address
TMR1_CTRL1	Control register 1	0x00
TMR1_CTRL2	Control register 2	0x04
TMR1_SMCTRL	Slave mode control register	0x08
TMR1_DIEN	DMA/Interrupt enable register	0x0C
TMR1_STS	Status register	0x10
TMR1_CEG	Control event generation register	0x14
TMR1_CCM1	Capture/Compare mode register 1	0x18

Register name	Description	Offset address
TMR1_CCM2	Capture/Compare mode register 2	0x1C
TMR1_CCEN	Capture/Compare enable register	0x20
TMR1_CNT	Counter register	0x24
TMR1_PSC	Prescale register	0x28
TMR1_AUTORLD	Auto reload register	0x2C
TMR1_REPCNT	Repeat count register	0x30
TMR1_CC1	Channel 1 capture/compare register	0x34
TMR1_CC2	Channel 2 capture/compare register	0x38
TMR1_CC3	Channel 3 capture/compare register	0x3C
TMR1_CC4	Channel 4 capture/compare register	0x40
TMR1_BDT	Braking and deadband register	0x44
TMR1_DCTRL	DMA control register	0x48
TMR1_DMADDR	DMA address register of continuous mode	0x4C

14.6 Register Functional Description

14.6.1 Control Register 1 (TMR1_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 write 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: Enable update event (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: Disable update event

Field	Name	R/W	Description
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 by this bit. 0: The counter overruns or underruns Set UEG bit Update generated by slave mode controller (An update interrupt or DMA request is generated by any of the above events) 1: The counter overruns or underruns (An update interrupt or DMA request is generated by only this event)
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 subsequent output level of the channel will no long 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 alignment modes affect the timing of setting the output comparison interrupt flag bit of the output channel to 1; when the counter is disabled (CNTEN=0), select the center alignment mode. 00: Edge-aligned mode 01: Center alignment mode 1 (the output compare interrupt flag bit of output channel is set to 1 when counting down) 10: Center alignment mode 2 (the output compare interrupt flag bit of output channel is set to 1 when counting up) 11: Center alignment mode 3 (the output compare interrupt flag bit of output channel is set to 1 when counting up/down)
7	ARPEN	R/W	Enable Auto-reload Preload of TMR1_AUTORLD Register When the buffer is disabled, modification of TMR1_AUTORLD by program will immediately lead to modification of the values loaded to the counter; when the buffer is enabled, modification of TMR1_AUTORLD by program will lead to modification of the values loaded to the counter at the next update event. 0: Disable 1: Enable

Field	Name	R/W	Description
9:8	CLKDIV	R/W	Clock Division During the configuration of deadband and digital filter, CK_INT provides the clock, and the deadband 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		

14.6.2 Control Register 2 (TMR1_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 the 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 It works only when the capture/compare preload is enabled (CCPEN=1), and 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: Transmit DMA request of CCx when CCx event occurs 1: Transmit DMA request of CCx when an update event occurs
6:4	MMSEL	R/W	Select Signal for TRGO in Timer Main Mode The signals of timers working in master mode can be used for TRGO, to affect the work of timers in slave mode and cascaded with the master timer, and the specific impact is related to the configuration of slave mode timer. 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

Field	Name	R/W	Description
7	TI1SEL	R/W	Timer Input 1 Select 0: TMR1_CH1 pin is connected to TI1 input 1: TMR1_CH1, TMR1_CH2 and TMR1_CH3 pins are connected to TI1 input after XOR operation
8	OC1OIS	R/W	OC1 Output Idle State Configure Only when MOEN=0 and OC1N is implemented, it only affects the level state after the deadband of OC1. 0: OC1=0 1: OC1=1 Note: When LOCKCFG bit in TMR1_BDT register is at Level 1, 2 or 3, this bit cannot be modified.
9	OC1NOIS	R/W	OC1N Output Idle State Configure Only the level state after the deadband of OC1 is affected when MOEN=0 and OC1N is realized. 0: OC1N=0 1: OC1N=1 Note: When LOCKCFG bit in TMR1_BDT register is at 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		

14.6.3 Slave Mode Control Register (TMR1_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 TI2FP2, the counter works at the edge of TI1FP1. 010: Encoder Mode 2; according to the level of TI1FP1, the counter works at the edge of TI2FP2. 011: Encoder mode 3; according to the input level of the other 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; when the slave mode timer receives the TRGI high level signal, the counter will start to work; when it receives TRGI low level signal, the counter will stop working; when it receives 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	OCCSEL	R/W	Select OCREF Clear Source This bit is used to select OCREF clear source 0: Reserved 1: ETRF
6:4	TRGSEL	R/W	Trigger Input Signal Select In order to avoid generating false edge detection when changing the value of this bit, it must be changed when SMFSEL=0. 000: Internal trigger ITR0 001: Internal trigger ITR1 010: Internal trigger ITR2 011: Reserved 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

Field	Name	R/W	Description
11:8	ETFCFG	R/W	External Trigger Filter Configure 0000: Disable filter, sampled 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. The clock frequency is t_{CK_INT} when DIV =1; for all other values, it is t_{DTS}.
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 TMR1CLK frequency; when ETR frequency is too high, the ETRP frequency must be reduced through frequency division. 00: Disable the prescaler 01: ETR signal 2 frequency division 10: ETR signal 4 frequency division 11: ETR signal 8 frequency division
14	ECEN	R/W	External Clock Enable Mode2 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 phase-inverting. 0: The external trigger ETR is not phase-inverting, and the high level or rising edge is valid 1: The external trigger ETR is phase-inverting, and the low level or falling edge is valid

Table 47 TMR1 Internal Trigger Connection

Slave timer	ITR0（TRGSEL=000）	ITR1（TRGSEL=001）	ITR2（TRGSEL=010）
TMR1	TMR4	TMR2	TMR3

14.6.4 DMA/Interrupt Enable Register (TMR1_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
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

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

14.6.5 Status register (TMR1_STS)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
0	UIFLG	RC_W0	Update Event Interrupt Generate Flag 0: No update event interrupt occurs 1: Update event interrupt occurred When the counter value is reloaded or reinitialized, an update event will be generated. The bit is set to 1 by hardware and cleared to 0 by software; update events are generated in the following situations: (1) UD=0 on TMR1_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 TMR1_CTRL1 register, configure UEG=1 on TMR1_CEG register to generate an update event, and the counter needs to be initialized by software; (3) URSSEL=0 and UD=0 on TMR1_CTRL1 register, and an update event will be generated when the counter is initialized by trigger event.
1	CC1IFLG	RC_W0	Capture/Compare Channel 1 Interrupt Flag When the capture/compare channel 1 is configured as output: 0: No matching occurs 1: The value of TMR1_CNT matches the value of TMR1_CC1 When the capture/compare channel 1 is configured as input: 0: No input capture occurs 1: Input capture occurs When a capture event occurs, this bit is set to 1 by hardware; and cleared by software or reading TMR1_CC1 register.
2	CC2IFLG	RC_W0	Capture/Compare Channel 2 Interrupt Flag Refer to the description of STS_CC1IFLG.
3	CC3IFLG	RC_W0	Capture/Compare Channel 3 Interrupt Flag Refer to the description of STS_CC1IFLG.
4	CC4IFLG	RC_W0	Capture/Compare Channel 4 Interrupt Flag Refer to the description of STS_CC1IFLG.

Field	Name	R/W	Description
5	COMIFLG	RC_W0	COM Event Interrupt Generate Flag 0: No COM event occurs 1: COM interrupt waits for response After COM event is generated, this bit is set to 1 by hardware and cleared to 0 by software.
6	TRGIFLG	RC_W0	Trigger Event Interrupt Generate Flag 0: No trigger event interrupt occurs 1: Trigger event interrupt occurs When a trigger event is generated, this bit is set to 1 by hardware and cleared to 0 by software.
7	BRKIFLG	RC_W0	Break Event Interrupt Generate Flag 0: No brake event occurs 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 to 0 by software.
8	Reserved		
9	CC1RCFLG	RC_W0	Capture/Compare Channel 1 Repetition Capture Flag 0: Repeated capture does not occur 1: Repeated capture occurs The value of the counter is captured to TMR1_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 Channel 2 Repetition Capture Flag Refer to the description of STS_CC1RCFLG.
11	CC3RCFLG	RC_W0	Capture/Compare Channel 3 Repetition Capture Flag Refer to the description of STS_CC1RCFLG.
12	CC4RCFLG	RC_W0	Capture/Compare Channel 4 Repetition Capture Flag Refer to the description of STS_CC1RCFLG.
15:13	Reserved		

14.6.6 Control Event Generation Register (TMR1_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 an update event This bit is set to 1 by software, and cleared to 0 by hardware. Note: When an update event is generated, the counter of the prescaler will be cleared to 0, but the prescaler factor remains unchanged. In the count-down mode, the counter reads the value of TMR1_AUTORLD; in center-aligned mode or count-up mode, the counter will be cleared.

Field	Name	R/W	Description
1	CC1EG	W	Capture/Compare Channel 1 Event Generation 0: Invalid 1: Generate capture/compare event This bit is set to 1 by software and cleared to 0 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 TMR1_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 Channel 2 Event Generation Refer to CC1EG description
3	CC3EG	W	Capture/Compare Channel 3 Event Generation Refer to CC1EG description
4	CC4EG	W	Capture/Compare Channel 4 Event Generation Refer to CC1EG description
5	COMG	W	Capture/Compare Control Update Event Generate 0: Invalid 1: Generate capture/Compare update event This bit is set to 1 by software and cleared to 0 automatically by hardware. Note: COMG bit is valid only in complementary output channel.
6	TEG	W	Trigger Event Generate 0: Invalid 1: Generate trigger event This bit is set to 1 by software and cleared to 0 automatically by hardware.
7	BEG	W	Break Event Generate 0: Invalid 1: Generate brake event This bit is set to 1 by software and cleared to 0 automatically by hardware.
15:8	Reserved		

14.6.7 Capture/Compare mode register 1 (TMR1_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 OCxx in the register describes the function of the channel in the output mode, and the ICxx 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 Select This bit defines the input/output direction and selects the 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 closed (TMR1_CCEN register CC1EN=0).
2	OC1FEN	R/W	Output Compare Channel 1 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 Channel 1 Preload Enable 0: Disable preloading function; write the value of TMR1_CC1 register through the program and it will work immediately. 1: Enable preloading function; write the value of TMR1_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 Channel 1 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 high 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 low 011: Output reverses when matching. When the value of the counter matches the value of the capture/compare register, reverse the level of OC1REF 100: The output is forced to be low. Force OC1REF to be low 101: The output is forced to be high. Force OC1REF to be high 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 Channel 1 Clear Enable 0: OC1REF is unaffected by ETRF input. 1: When high level of ETRF input is detected, OC1REF=0

Field	Name	R/W	Description
9:8	CC2SEL	R/W	Capture/Compare Channel 2 Select This bit defines the input/output direction and selects the 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 closed (TMR1_CCEN register CC2EN=0).
10	OC2FEN	R/W	Output Compare Channel 2 Fast Enable
11	OC2PEN	R/W	Output Compare Channel 2 Preload Enable
14:12	OC2MOD	R/W	Output Compare Channel 2 Mode
15	OC2CEN	R/W	Output Compare Channel 2 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 closed (TMR1_CCEN bit CC1EN=0).
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 prescaler factor; capture is triggered once by every PSC events.

Field	Name	R/W	Description
7:4	IC1F	R/W	Input Capture Channel 1 Filter Configure 0000: Disable filter, sampled 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. The clock frequency is t_{CK_INT} when DIV =1; for all other values, it is t_{DTS}.
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 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 closed (TMR1_CCEN register CC2EN=0).
11:10	IC2PSC	R/W	Input Capture Channel 2 Perscaler Configure
15:12	IC2F	R/W	Input Capture Channel 2 Filter Configure

14.6.8 Capture/Compare mode register 2 (TMR1_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 3 Select This bit defines the input/output direction and selects the 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 Note: This bit can be written only when the channel is closed (TMR1_CCEN register CC3EN=0).

Field	Name	R/W	Description
2	OC3FEN	R/W	Output Compare Channel 3 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 Channel 3 Preload Enable
6:4	OC3MOD	R/W	Output Compare Channel 3 Mode Configure
7	OC3CEN	R/W	Output Compare Channel 3 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 Select This bit defines the input/output direction and selects the 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 closed (TMR1_CCEN register CC4EN=0).
10	OC4FEN	R/W	Output Compare Channel 4 Fast Enable
11	OC4PEN	R/W	Output Compare Channel 4 Buffer Enable
14:12	OC4MOD	R/W	Output Compare Channel 4 Mode Configure
15	OC4CEN	R/W	Output Compare Channel 4 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 closed (TMR1_CCEN register CC3EN=0).
3:2	IC3PSC	R/W	Input Capture Channel 3 Perscaler Configure 00: PSC=1 01: PSC=2 10: PSC=4 11: PSC=8 PSC is prescaler factor; capture is triggered once by every PSC events.
7:4	IC3F	R/W	Input Capture Channel 3 Filter Configure

Field	Name	R/W	Description
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 closed (TMR1_CCEN register CC4EN=0).
11:10	IC4PSC	R/W	Input Capture Channel 4 Perscaler Configure
15:12	IC4F	R/W	Input Capture Channel 4 Filter Configure

14.6.9 Capture/Compare enable register (TMR1_CCEN)

Offset address: 0x20

Reset value: 0x0000

Field	Name	R/W	Description
0	CC1EN	R/W	Capture/Compare Channel 1 Output Enable When CC1 is configured as output: 0: Disable output 1: Enable output When CC1 is configured as input: This bit determines whether the value CNT of the counter can be captured and enter TMR1_CC1 register 0: Disable capture 1: Enable capture
1	CC1POL	R/W	Capture/Compare Channel 1 Output Polarity Configure When CC1 channel is configured as output: 0: OC1 is active high 1: OC1 is active low When CC1 channel is configured as input: CC1POL and CC1NPOL control the polarity of the triggered or captured signals TI1FP1 and TI2FP1 at the same time. 00: Non-phase-inverting/rising edge: TIxFP1 is not phase-inverting (triggered in gated and encoder mode), and is captured at the rising edge of TixFP1 (reset trigger, capture, external clock and trigger mode). 01: Phase inverting/falling edge: TixFP1 is phase-inverting (triggered in gated and encoder mode), and is captured at the falling edge of TixFP1 (reset trigger, capture, external clock and trigger mode). 10: Reserved 11: Non-phase-inverting/Rising and falling edges: TixFP1 is not phase-inverting (triggered in gated mode, cannot be used in encoder mode), and is captured at the rising and falling edge of TixFP1 (reset trigger, capture, external clock and trigger mode).
2	CC1NEN	R/W	Enable Capture/Compare Channel 1 Complementary Output 0: Disable 1: Enable

Field	Name	R/W	Description
3	CC1NPOL	R/W	Capture/Compare Channel 1 Complementary Output Polarity When CC1 channel is configured as output 0: OC1N is active high 1: OC1N is active low When CC1 channel is configured as input This bit, together with CC1POL, defines the polarity of TI1FP1 and TI2FP1 Note: On the complementary output channel, if this bit is preloaded, and CCPEN=1 for TMR1_CTRL2, CC1NPOL can obtain new value from the preload bit only when a reversing event is generated. When the protection level is 2 or 3, this bit cannot be modified
4	CC2EN	R/W	Capture/Compare Channel 2 Output Enable Refer to CCEN_CC1EN
5	CC2POL	R/W	Capture/Compare Channel 2 Output Polarity Configure Refer to CCEN_CC1POL
6	CC2NEN	R/W	Enable Capture/Compare Channel 1 Complementary Output Refer to CCEN_CC1NEN
7	CC2NPOL	R/W	Configure Complementary Output Polarity of Capture/Compare Channel 2 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	Enable Capture/Compare Channel 3 Complementary Output Refer to CCEN_CC1NEN
11	CC3NPOL	R/W	Configure Complementary Output Polarity of Capture/Compare Channel 3 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
14	Reserved		
15	CC4NPOL	R/W	Configure Complementary Output Polarity of Capture/Compare Channel 4 Refer to CCEN_CC1NPOL

14.6.10 Counter Register (TMR1_CNT)

Offset address: 0x24

Reset value: 0x0000

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

14.6.11 Prescaler Register (TMR1_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)

14.6.12 Auto Reload Register (TMR1_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.

14.6.13 Repeat Count Register (TMR1_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		

14.6.14 Capture/Compare Channel 1 register (TMR1_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 Capture/Compare Channel 1 is configured as output mode: CC1 contains the value currently loaded in the capture/compare register. Compare the value CC1 of the capture and 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 TMR1_CCM1 register), the written value will immediately affect the output comparison results; If the output compare preload is enabled (OC1PEN=1 for TMR1_CCM1 register), the written value will affect the output comparison result when an update event is generated.

14.6.15 Capture/Compare Channel 2 register (TMR1_CC2)

Offset address: 0x38

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC2	R/W	Capture/Compare Channel 2 Value Refer to TMR1_CC1

14.6.16 Capture/Compare Channel 3 register (TMR1_CC3)

Offset address: 0x3C

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC3	R/W	Capture/Compare Channel 3 Value Refer to TMR1_CC1

14.6.17 Capture/Compare Channel 4 register (TMR1_CC4)

Offset address: 0x40

Reset value: 0x0000

Field	Name	R/W	Description
15:0	CC4	R/W	Capture/Compare Channel 4 Value Refer to TMR1_CC1

14.6.18 Brake and Deadband Register (TMR1_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 TMR1_BDT register for the first time.

When performing consecutive write operations to this register, confirm each write before proceeding with the next write operation.

Field	Name	R/W	Description
7:0	DTS	R/W	Deadband Setup DT is the deadband duration, and the relationship between DT and register DTS is as follows: DTS[7:5]=0xx=>DT=DTS[7:0]×T_{dts}, T_{dts}=T_{DTS}; 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}=125 ns (8 MHz), the deadband is set as follows: If the step time is 125ns, the deadband can be set from 0 to 15875ns; If the step time is 250ns, the deadband can be set from 16us to 31750ns; If the step time is 1 μs, the deadband can be set from 32 μs to 63 μs; If the step time is 2 μs, the deadband can be set from 64 μs to 126 μs. Note: Once LOCK level (LOCKCFG bit in TMR1_BDT register) is set to 1, 2 or 3, these bits cannot be modified.

Field	Name	R/W	Description
9:8	LOCKCFG	R/W	Lock Write Protection Mode Configure 00: No Lock write protection; it can be written to the register directly 01: Lock write protection level 1 It cannot be written to DTS, BRKEN, BRKPOL and AOEN bits of TMR1_BDT, and OCxOIS and OCxNOIS bits of TMR1_CTRL2 register. 10: Lock write protection level 2 It cannot be written to all bits at Protection Level 1, CCxPOL and OCxNPOL bits in TMR1_CCEN register, and RMOS and IMOS bits in TMR1_BDT register. 11: Lock write protection level 3 It cannot be written to all bits at Protection Level 2, and OCxMOD and OCxPEN bits of TMR1_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; disable 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: Disable Ocx/OcxN output 1: If CcxEN=1, the invalid level is output during the deadband (the specific level value is affected by the polarity configuration), and the idle level is output after the deadband
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. 0: Disable Ocx/OcxN output 1: Ocx/OcxN first outptus invalid level (the specific level value is affected by the polarity configuration)
12	BRKEN	R/W	Break Function Enable 0: Disable 1: Enable Note: When the protection level is 1, this bit cannot be modified. Writing to this bit requires an APB clock delay before use.
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 use.
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 at the next update event (braking input is invalid) Note: When the protection level is 1, this bit cannot be modified.

Field	Name	R/W	Description
15	MOEN	R/W	Enable Waveform Main Output 0: Disable the output of Ocx and OcxN or force the output of idle state 1: When CcxEN and CCxNEN bits of the TMR1_CCEN register are set, Ocx and OcxN outputs are enabled When the brake input is valid, it is cleared to 0 by hardware asynchronously. Note: Setting 1 by software or setting 1 automatically depends on AOEN bit of the TMR1_BDT register.

14.6.19 DMA Control Register (TMR1_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 TMR1_DMA register), and DBADDR is defined as the offset from the address of TMR1_CTRL1 register: 00000：TMR1_CTRL1 00001：TMR1_CTRL2 00010：TMR1_SMCTRL …….
7:5	Reserved		
12:8	DBLEN	R/W	DMA Burst Transfer Length Setup These bits define the transmission length and transmission times of DMA in continuous mode. The data transmitted can be 16 bits and 8 bits. When reading/writing TMR1_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=TMR1_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 TMR1_CTRL1 address +DBADDR+7 means the address of the data to be written/read. Data transmission will occur to: TMR1_CTRL1 address + seven registers starting from DBADDR. The data transmission will change according to different DMA data length: When the transmission data is set to 16 bits, the data will be transmitted to seven registers. 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		

14.6.20 DMA Address Register of Continuous Mode (TMR1_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 TMR1_DMADDR register may lead to access to the register in the following address: TMR1_CTRL1 address + (DBADDR+DMA index) ×4 Wherein: “TMR1_CTRL1 address” is the address of control register 1 (TMR1_CTRL1); “DBADDR” is the base address defined in TMR1_DCTRL register; “DMA index” is the offset automatically controlled by DMA, and it depends on DBLEN defined in TMR1_DCTRL register.

15 General-Purpose Timer (TMR2/3/4)

15.1 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. Include a 16-bit auto reload counter (realize up, down and center-aligned counting).

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

15.2 Main Characteristics

(1) Timebase unit
- Counter: 16-bit counter, supporting count-up, count-down and center-aligned count
- Prescaler: 16-bit programmable prescaler
- Autoreload function

(2) Clock source selection
- Internal clock
- External input
- Internal trigger

(3) Input function
- Counting function
- PWM input
- Encoder interface mode

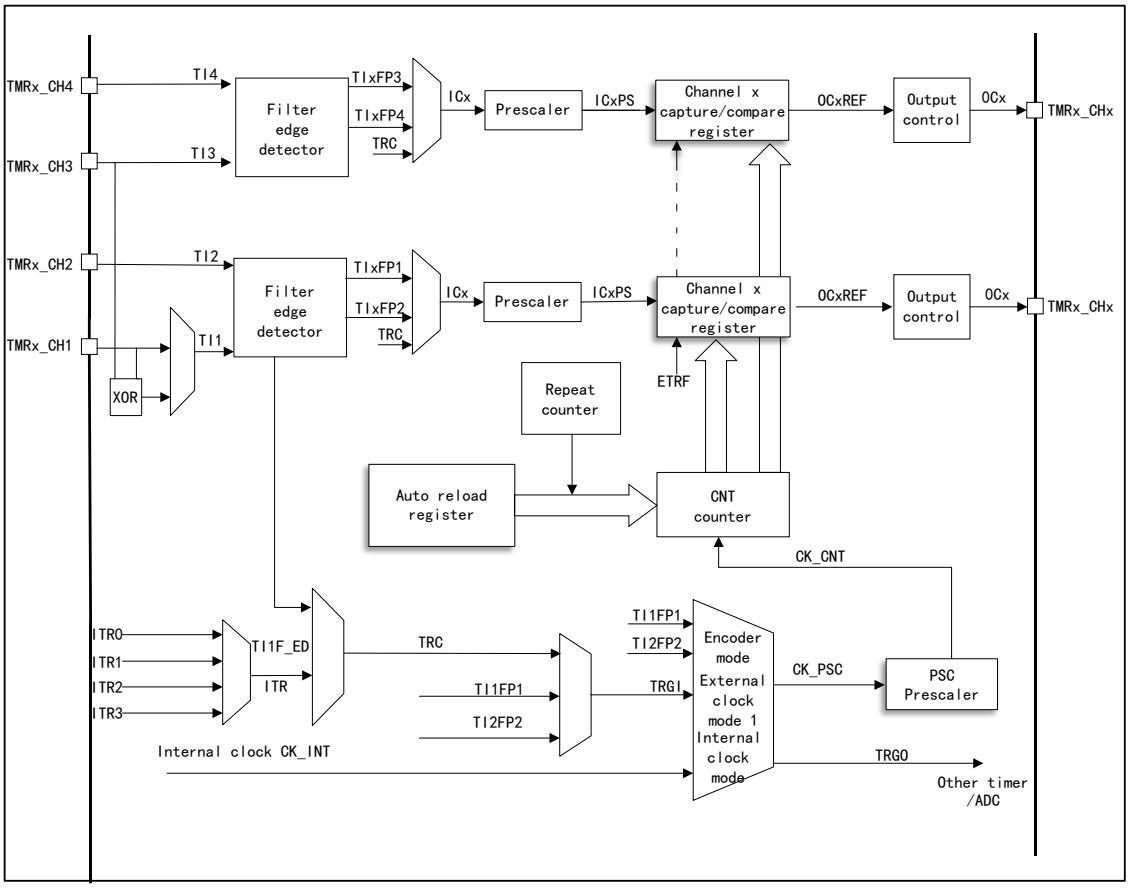
(4) Output function
- PWM output mode
- Forced output mode
- Single-pulse mode

(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

15.3 Structure Block Diagram

Figure 44 General-Purpose Timer TMR2/3/4 Structure Block Diagram



15.4 Functional Description

15.4.1 Clock source selection

The general-purpose timer has three 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 input channels TI1/2/3/4 from the timer itself generate trigger signals after polarity selection and filtering, which are then connected to the slave mode controller to control the operation of the counter. Specifically, the input on channel 1 undergoes both rising and falling edge detection, and the resulting

pulse signals are logically Ored to form the TI1F_ED signal, also known as the TIF_ED dual-edge signal. Notably, PWM input can only be provided by TI1 or TI2.

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 general-purpose timer contains three registers

- Counter register (CNT) 16 bits
- Autoreload 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

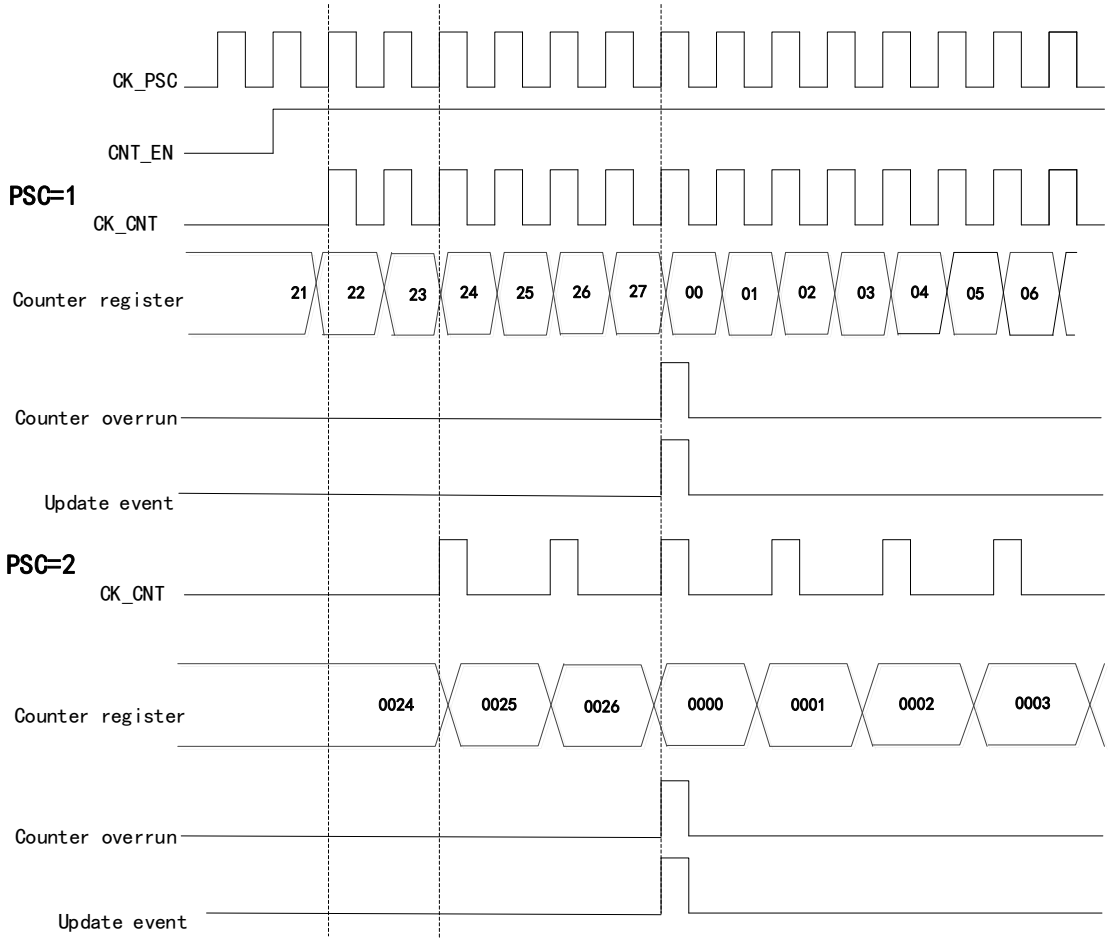
Count-up mode

Set to the count-up mode by configuring CNTDIR bit of 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 from 0 again, 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 configuring UD bit of control register TMRx_CTRL1.

Figure 45 Timing Diagram of Count-up Mode when Division Factor is 1 or 2



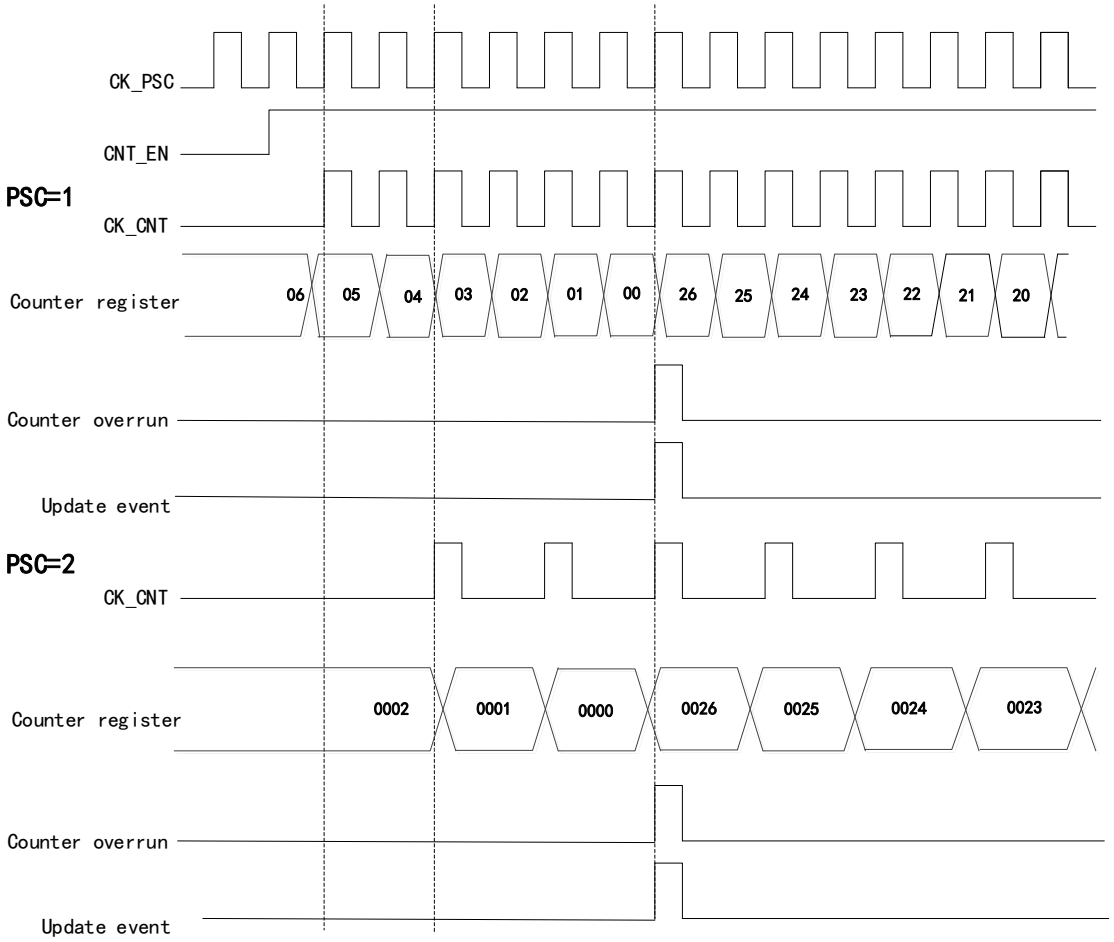
Count-down mode

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

When the counter is in count-down mode, it 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 46 Timing Diagram of Count-down Mode when Division Factor is 1 or 2

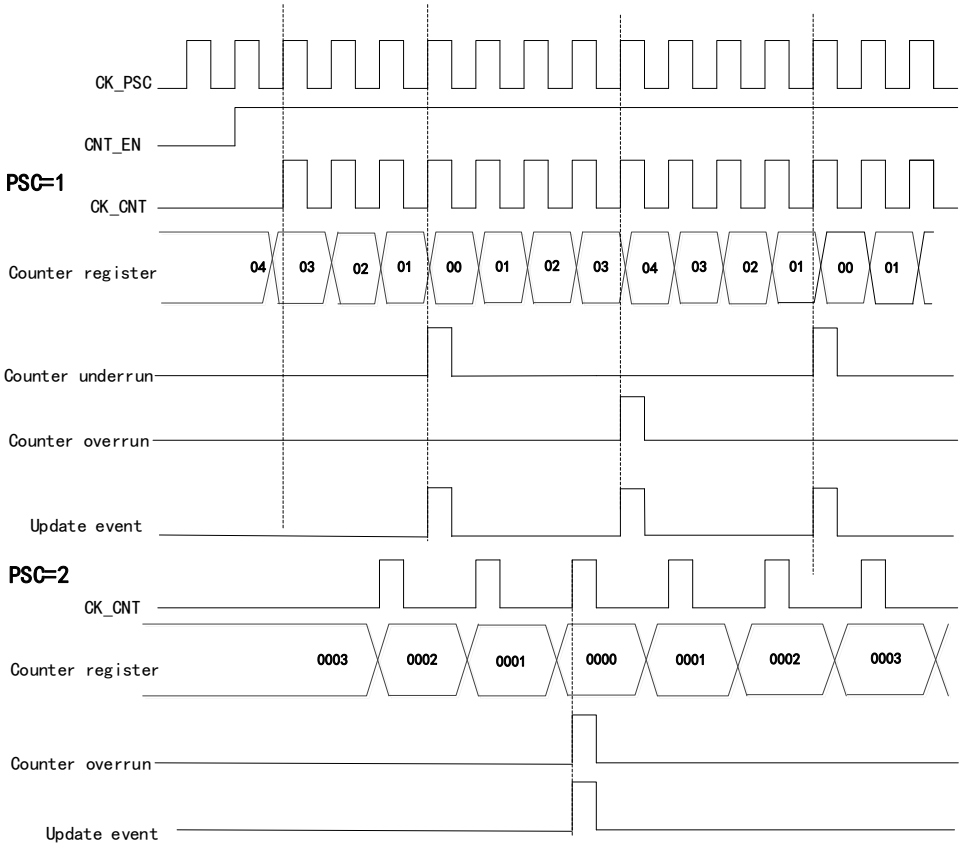


Center-aligned mode

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

When the counter is in center-aligned mode, the counter counts up from 0 and when it reaches the value of auto reload (TMRx_AUTORLD), it 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 47 Timing Diagram of Center-Aligned Mode when 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 within 1~65536 (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 general-purpose 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 enter the capture channels. 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 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 status register TMRx_STS will be set to 1; if CCxIEN=1, an interrupt will be generated.

In capture mode, the timing, frequency, cycle 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; read the value of capture register and the cycle of this pulse signal will be obtained by capture.

15.4.4 Output compare

There are eight modes of output compare: freeze, channel x is valid when matching, channel x is invalid when matching, flip, force to invalid, force to valid, PWM mode 1 and PWM mode 2, which are configured by OCxMOD bit in TMR_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, a DMA request will be generated.

15.4.5 PWM output mode

PWM mode is pulse signal that can be adjusted by external output of 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 PWM Mode 1 when CCx=5, AUTORLD=7:

Figure 48 Timing Diagram of PWM1 Count-up Mode

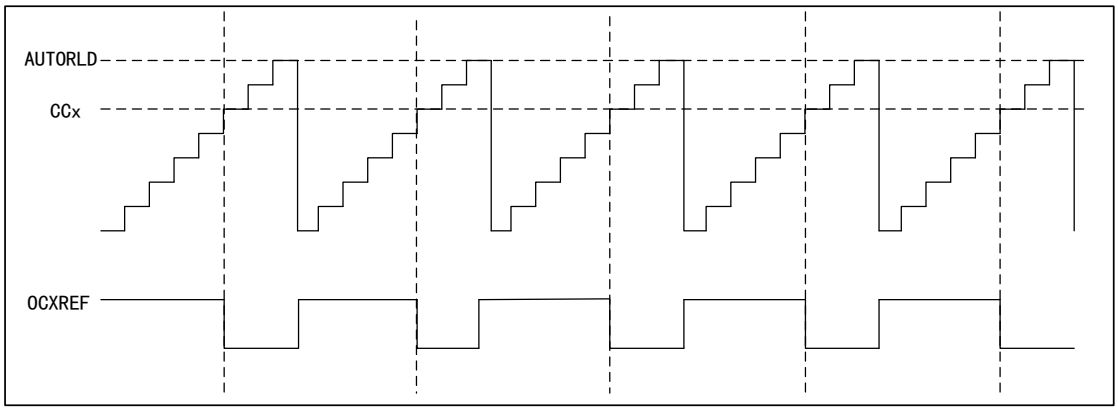


Figure 49 Timing Diagram of PWM1 Count-down Mode

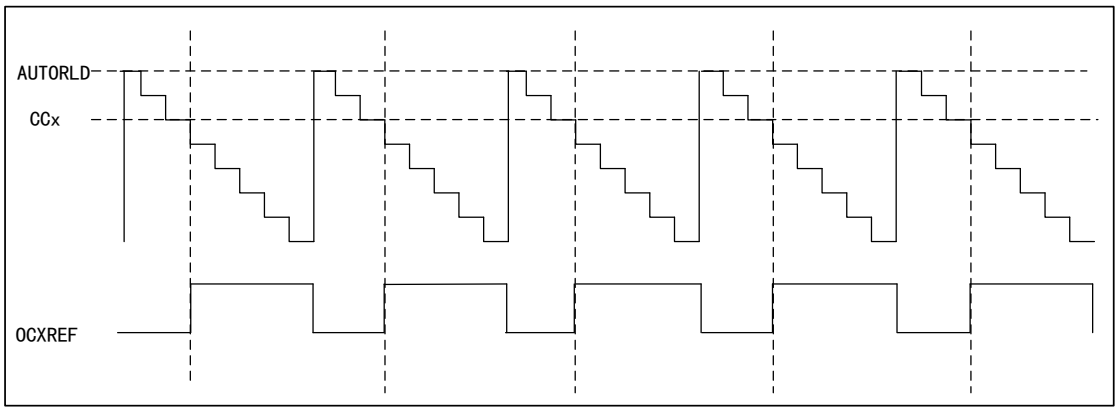
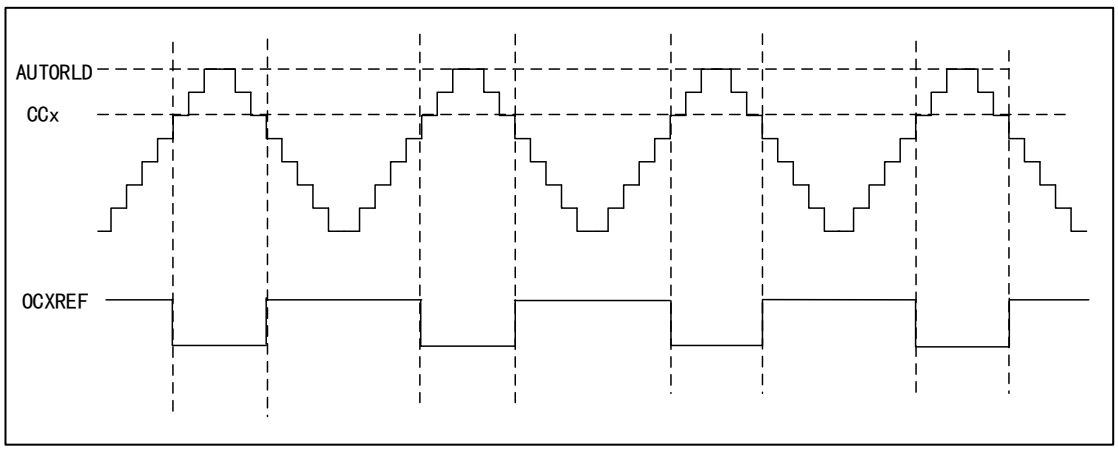


Figure 50 Timing Diagram of PWM1 Center-Aligned Mode



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 51 Timing Diagram of PWM2 Count-up Mode

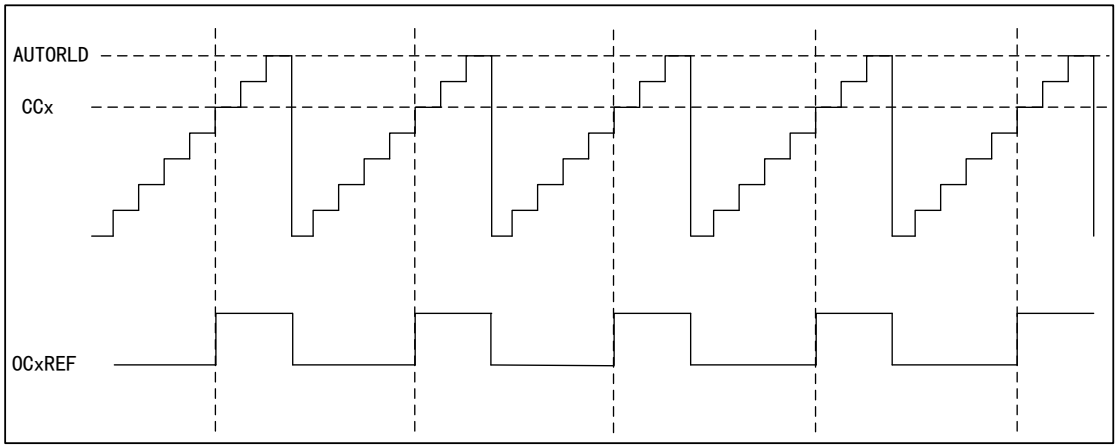


Figure 52 Timing Diagram of PWM2 Count-down Mode

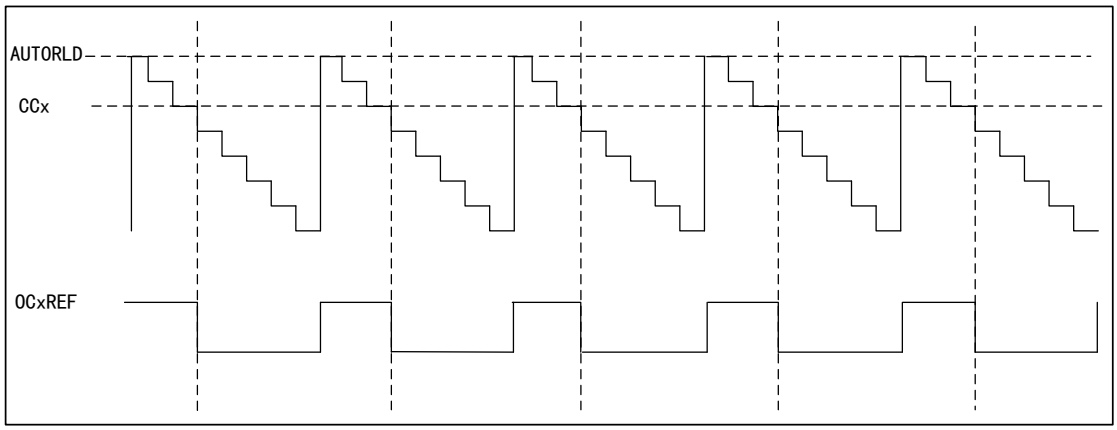
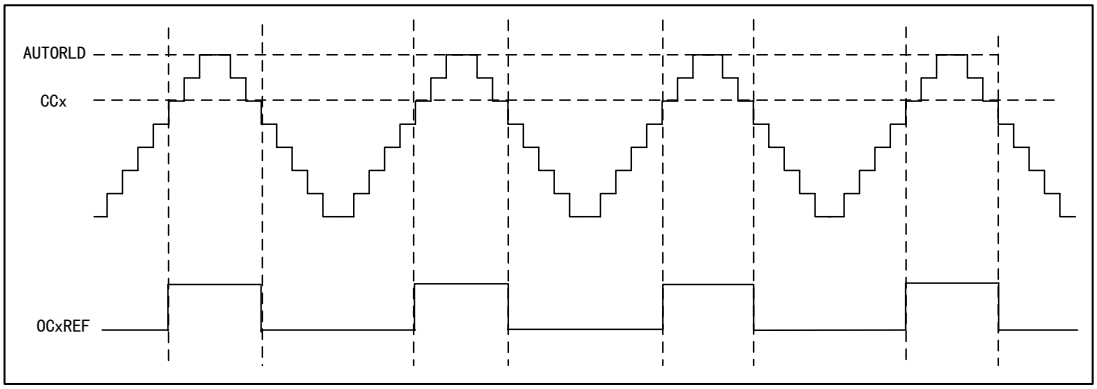


Figure 53 Timing Diagram of PWM2 Center-Aligned Mode



15.4.6 PWM input mode

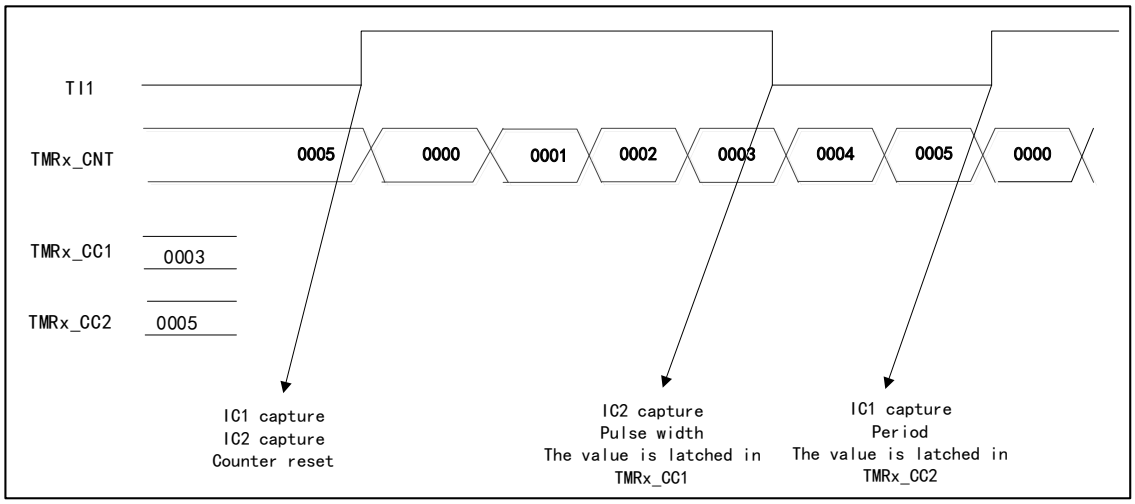
PWM input mode is a particular case of input capture.

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 capture 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 54 Timing Diagram in PWM Input Mode



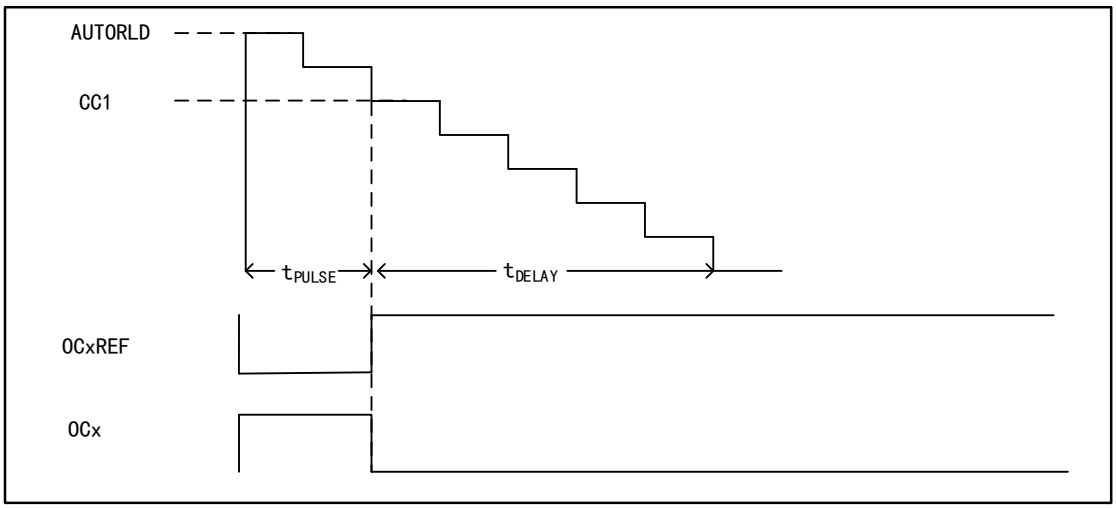
15.4.7 Single-pulse mode

The single-pulse mode is a special case of timer compare 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 55 Timing Diagram of Single-Pulse Mode



15.4.8 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

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

15.4.9 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 method of selecting 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 48 Relationship between Count Direction and Encoder

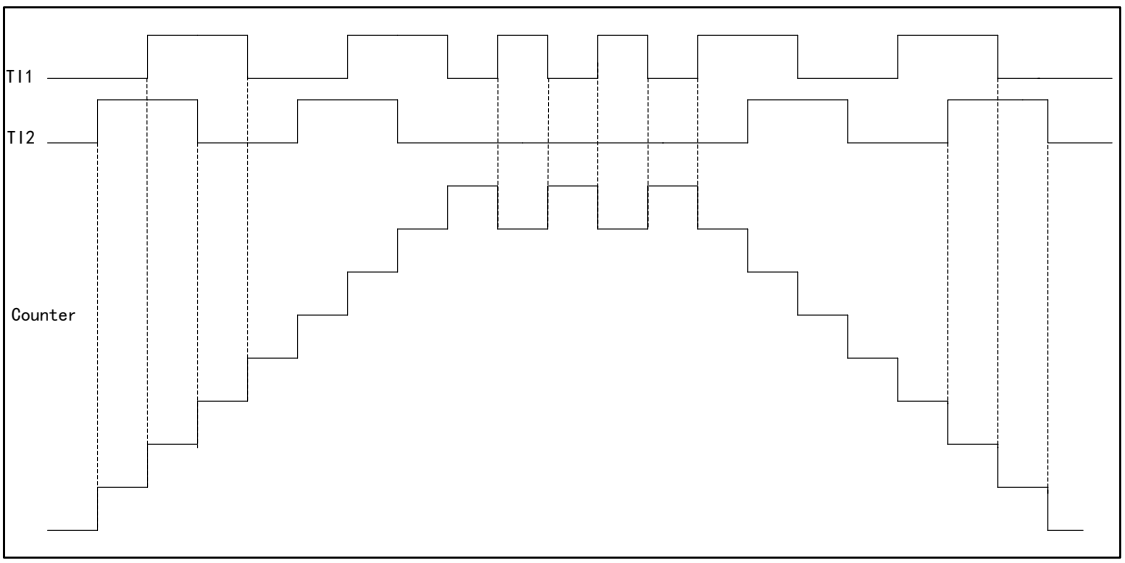
Effective edge		**Count only in TI1**		**Count only in TI2**		**Count in both TI1 and TI2**	
Level of relative signal		High	Low	High	Low	High	Low
TI1FP1	Rising Edge	—		Count down	Count up	Count down	Count up
	Falling Edge			Count up	Count down	Count up	Count down
TI2FP2	Rising Edge	Count up	Count down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

Among the following examples:

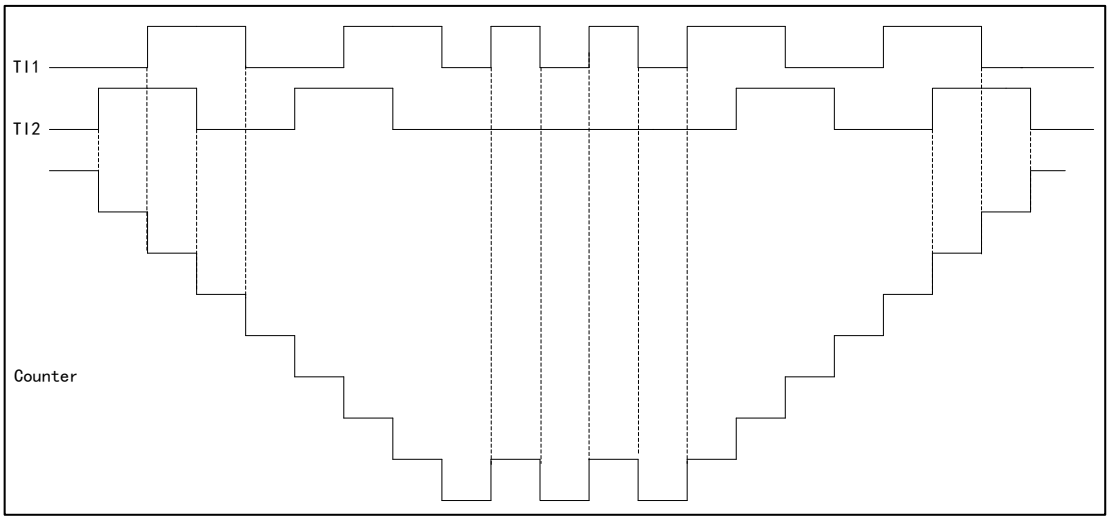
- IC1FP1 is mapped to TI1
- IC2FP2 is mapped to TI2
- Neither IC1FP1 nor IC2FP2 is phase-inverting
- The input signal is valid at the rising edge and falling edge
- Enable the counter

Figure 56 Operation Instance of Counter in Encoder Mode



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

Figure 57 Encoder Interface Mode Instance with IC1FP1 Inverted



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

15.4.10 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, and 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 end. When the trigger input is high, the clock of the counter will be enabled. 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 will be enabled at the rising edge of the trigger input (but not be reset), and only the start of the counter is controlled.

15.4.11 Timer interconnection

Refer to the section “Timer Interconnection” of TMR1 for details.

15.4.12 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 trigger)
- Capture/Compare event

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

15.4.13 Clock function test

TMR3 can map different clock sources to CH2 by configuring the OPT register, and the frequency of this clock can then be measured by utilizing the input capture function.

15.5 Register Address Mapping

In the following table, all registers of TMR3/TMR4 are mapped to a 16-bit addressable (addressing) space.

Table 49 TMR3 and TMR4 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	Status 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
TMRx_CNT	Counter register	0x24
TMRx_PSC	Prescale register	0x28
TMRx_AUTORLD	Auto load register	0x2C
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_DCTRL	DMA control register	0x48

Register name	Description	Offset address
TMRx_DMADDR	DMA address register of continuous mode	0x4C
TMR3_OPT	Input Channel Remapping Register	0x50

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 write 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: Enable update event (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: Disable update event
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 by 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 subsequent output level of the channel will no long 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

Field	Name	R/W	Description
6:5	CAMSEL	R/W	Select Center-Aligned Mode In the center-aligned mode, the counter counts up and down alternately; otherwise, it will only count up or down. Different center alignment modes affect the timing of setting the output comparison interrupt flag bit of the output channel to 1; when the counter is disabled (CNTEN=0), select the center alignment mode. 00: Edge-aligned mode 01: Center alignment mode 1 (the output compare interrupt flag bit of output channel is set to 1 when counting down) 10: Center alignment mode 2 (the output compare interrupt flag bit of output channel is set to 1 when counting up) 11: Center alignment mode 3 (the output compare interrupt flag bit of output channel is set to 1 when counting up/down)
7	ARPEN	R/W	Enable Auto-reload Preload of TMRx_AUTORLD Register When the buffer is disabled, modification of TMRx_AUTORLD by program will immediately lead to modification of the values loaded to the counter; when the buffer is enabled, modification of TMRx_AUTORLD by program will lead to modification of the values loaded to the counter at the next update event. 0: Disable 1: Enable
9:8	CLKDIV	R/W	Clock Division During the configuration of deadband and digital filter, CK_INT provides the clock, and the deadband 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
2:0	Reserved		
3	CCDSEL	R/W	Select Sending Capture/Compare DMA Request 0: Transmit DMA request of CCx when CCx event occurs 1: Transmit DMA request of CCx when an update event occurs

Field	Name	R/W	Description
6:4	MMSEL	R/W	Master Mode Signal Select The signals of timers working in master mode can be used for TRGO, to affect the work of timers in slave mode and cascaded with the master timer, and the specific impact is related to the configuration of slave mode timer. 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 Select 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		

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 TI2FP2, the counter works at the edge of TI1FP1. 010: Encoder Mode 2; according to the level of TI1FP1, the counter works at the edge of TI2FP2. 011: Encoder mode 3; according to the input level of the other 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; when the slave mode timer receives the TRGI high level signal, the counter will start to work; when it receives TRGI low level signal, the counter will stop working; when it receives 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		

Field	Name	R/W	Description
6:4	TRGSEL	R/W	Trigger Input Signal Select In order to avoid generating false edge detection when changing the value of this bit, it must be changed when SMFSEL=0. 000: Internal trigger ITR0 001: Internal trigger ITR1 010: Internal trigger ITR2 011: Reserved 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: Reserved
7	MSMEN	R/W	Master/slave Mode Enable 0: Invalid 1: Enable the master/slave mode
15:8	Reserved		

Table 50 TMRx Internal Trigger Connection

Slave timer	ITR0（TRGSEL=000）	ITR1（TRGSEL=001）	ITR2（TRGSEL=010）
TMR2	TMR1	TMR3	TMR4
TMR3	TMR1	TMR2	TMR4
TMR4	TMR1	TMR2	TMR3

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 Channel 1 Interrupt Enable 0: Disable 1: Enable
2	CC2IEN	R/W	Capture/Compare Channel 2 Interrupt Enable 0: Disable 1: Enable
3	CC3IEN	R/W	Capture/Compare Channel 3 Interrupt Enable 0: Disable 1: Enable
4	CC4IEN	R/W	Capture/Compare Channel 4 Interrupt Enable 0: Disable 1: Enable
5	Reserved		

Field	Name	R/W	Description
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 Channel 1 DMA Request Enable 0: Disable 1: Enable
10	CC2DEN	R/W	Capture/Compare Channel 2 DMA Request Enable 0: Disable 1: Enable
11	CC3DEN	R/W	Capture/Compare Channel 3 DMA Request Enable 0: Disable 1: Enable
12	CC4DEN	R/W	Capture/Compare Channel 4 DMA Request Enable 0: Disable 1: Enable
13	Reserved		
14	TRGDEN	R/W	Trigger DMA Request Enable 0: Disable 1: Enable
15	Reserved		

15.6.5 Status register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
0	UIFLG	RC_W0	Update Event Interrupt Generate Flag 0: No update event interrupt occurs 1: Update event interrupt occurred When the counter value is reloaded or reinitialized, an update event will be generated. The bit is set to 1 by hardware and cleared to 0 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 an update event, and the counter needs to be initialized by software; (3) URSSEL=0 and UD=0 on TMRx_CTRL1 register, and an update event will be generated when the counter is initialized by trigger event.

Field	Name	R/W	Description
1	CC1IFLG	RC_W0	Capture/Compare Channel 1 Interrupt Flag When the capture/compare channel 1 is configured as output: 0: No matching occurs 1: The value of TMRx_CNT matches the value of TMRx_CC1 When the capture/compare channel 1 is configured as input: 0: No input capture occurs 1: Input capture occurs When a capture event occurs, set 1 by hardware; clear 0 by software or clear 0 when reading TMRx_CC1 register.
2	CC2IFLG	RC_W0	Capture/Compare Channel 2 Interrupt Flag Refer to the description of STS_CC1IFLG.
3	CC3IFLG	RC_W0	Capture/Compare Channel 3 Interrupt Flag Refer to the description of STS_CC1IFLG.
4	CC4IFLG	RC_W0	Capture/Compare Channel 4 Interrupt Flag Refer to the description of STS_CC1IFLG.
5	Reserved		
6	TRGIFLG	RC_W0	Trigger Event Interrupt Generate Flag 0: No trigger event interrupt occurs 1: Trigger event interrupt occurs When a trigger event is generated, this bit is set to 1 by hardware and cleared to 0 by software.
8:7	Reserved		
9	CC1RCFLG	RC_W0	Capture/Compare Channel 1 Repetition Capture Flag 0: Repeated capture does not occur 1: Repeated capture occurs The value of the counter is captured to TMRx_CCR1 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 Channel 2 Repetition Capture Flag Refer to the description of STS_CC1RCFLG.
11	CC3RCFLG	RC_W0	Capture/Compare Channel 3 Repetition Capture Flag Refer to the description of STS_CC1RCFLG.
12	CC4RCFLG	RC_W0	Capture/Compare Channel 4 Repetition Capture Flag Refer to the description of 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 an update event This bit is set to 1 by software, and cleared to 0 by hardware. Note: When an update event is generated, the counter of the prescaler will be cleared to 0, 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 Channel 1 Event Generation 0: Invalid 1: Generate capture/compare event This bit is set to 1 by software and cleared to 0 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 Channel 2 Event Generation Refer to CC1EG description
3	CC3EG	W	Capture/Compare Channel 3 Event Generation Refer to CC1EG description
4	CC4EG	W	Capture/Compare Channel 4 Event Generation Refer to CC1EG description
5	Reserved		
6	TEG	W	Trigger Event Generate 0: Invalid 1: Generate trigger event This bit is set to 1 by software and cleared to 0 automatically by hardware.
15:7	Reserved		

15.6.7 Capture/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 Ocxx in the register describes the function of the channel in the output mode, and the Icxx 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 Select This bit defines the input/output direction and selects the 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 closed (TMRx_CCEN register CC1EN=0).
2	OC1FEN	R/W	Output Compare Channel 1 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 Channel 1 Preload Enable 0: Disable preloading function; write the value of TMRx_CC1 register through the program and it will work immediately. 1: Enable preloading function; 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 Channel 1 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 high 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 low 011: Output reverses when matching. When the value of the counter matches the value of the capture/compare register, reverse the level of OC1REF 100: The output is forced to be low. Force OC1REF to be low 101: The output is forced to be high. Force OC1REF to be high 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	Reserved		

Field	Name	R/W	Description
9:8	CC2SEL	R/W	Capture/Compare Channel 2 Select This bit defines the input/output direction and selects the 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 closed (TMRx_CCEN register CC2EN=0).
10	OC2FEN	R/W	Output Compare Channel 2 Fast Enable
11	OC2PEN	R/W	Output Compare Channel 2 Preload Enable
14:12	OC2MOD	R/W	Output Compare Channel 2 Mode
15	Reserved		

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 closed (TMRx_CCEN bit CC1EN=0).
3:2	IC1PSC	R/W	Input Capture Channel1 Perscaler Configure 00: PSC=1 01: PSC=2 10: PSC=4 11: PSC=8 PSC is prescaler factor; capture is triggered once by every PSC events.

Field	Name	R/W	Description
7:4	IC1F	R/W	Input Capture Channel 1 Filter Configure 0000: Disable filter, sampled 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. The clock frequency is t_{CK_INT} when DIV =1; for all other values, it is T_{DTS}.
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 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 closed (TMRx_CCEN register CC2EN=0).
11:10	IC2PSC	R/W	Input Capture Channel 2 Perscaler Configure
15:12	IC2F	R/W	Input Capture Channel 2 Filter Configure

15.6.8 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 3 Select This bit defines the input/output direction and selects the 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 Note: This bit can be written only when the channel is closed (TMRx_CCEN register CC3EN=0).

Field	Name	R/W	Description
2	OC3FEN	R/W	Rapid Enable Output Compare Channel 3 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	Reserved		
9:8	CC4SEL	R/W	Capture/Compare Channel 4 Select This bit defines the input/output direction and selects the 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 closed (TMRx_CCEN register CC4EN=0).
10	OC4FEN	R/W	Output Compare Channel4 Fast Enable
11	OC4PEN	R/W	Output Compare Channel4 Buffer Enable
14:12	OC4MOD	R/W	Output Compare Channel4 Mode Configure
15	Reserved		

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 closed (TMRx_CCEN register CC3EN=0).
3:2	IC3PSC	R/W	Input Capture Channel3 Perscaler Configure 00: PSC=1 01: PSC=2 10: PSC=4 11: PSC=8 PSC is prescaler factor; capture is triggered once by every PSC events.
7:4	IC3F	R/W	Input Capture Channel3 Filter Configure

Field	Name	R/W	Description
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 closed (TMRx_CCEN register CC4EN=0).
11:10	IC4PSC	R/W	Input Capture Channel4 Perscaler Configure
15:12	IC4F	R/W	Input Capture Channel4 Filter Configure

15.6.9 Capture/Compare 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 CC1 is configured as output: 0: Disable output 1: Enable output When CC1 is configured as input: This bit determines whether the value CNT of the counter can be captured and enter TMRx_CC1 register 0: Disable capture 1: Enable capture
1	CC1POL	R/W	Capture/Compare Channel1 Output Polarity Configure When CC1 channel is configured as output: 0: OC1 is active high 1: OC1 is active low When CC1 channel is configured as input: CC1POL and CC1NPOL control the polarity of the triggered or captured signals TI1FP1 and TI2FP1 at the same time. 00: Non-phase-inverting/rising edge: TixFP1 is not phase-inverting (triggered in gated and encoder mode), and is captured at the rising edge of TixFP1 (reset trigger, capture, external clock and trigger mode). 01: Phase inverting/falling edge: TixFP1 is phase-inverting (triggered in gated and encoder mode), and is captured at the falling edge of TixFP1 (reset trigger, capture, external clock and trigger mode). 10: Reserved 11: Non-phase-inverting/Rising and falling edges: TixFP1 is not phase-inverting (triggered in gated mode, cannot be used in encoder mode), and is captured at the rising and falling edge of TixFP1 (reset trigger, capture, external clock and trigger mode).
2	Reserved		

Field	Name	R/W	Description
3	CC1NPOL	R/W	Capture/Compare Channel 1 Output Polarity Configure When CC1 channel is configured as output: CC1NPOL remains in cleared state all the time When CC1 channel is configured as input: This bit and CC1POL control the polarity of the triggered or captured signals TI1FP1 and TI2FP1 at the same time.
4	CC2EN	R/W	Capture/Compare Channel 2 Output Enable Refer to CCEN_CC1EN
5	CC2POL	R/W	Capture/Compare Channel 2 Output Polarity Configure Refer to the description of CCEN_CC1POL.
6	Reserved		
7	CC2NPOL	R/W	Configure Output Polarity of Capture/Compare Channel 2 Refer to the description of CCEN_CC1NPOL.
8	CC3EN	R/W	Capture/Compare Channel 3 Output Enable Refer to the description of CCEN_CC1EN.
9	CC3POL	R/W	Capture/Compare Channel 3 Output Polarity Configure Refer to the description of CCEN_CC1POL.
10	Reserved		
11	CC3NPOL	R/W	Capture/Compare Channel 3 Output Polarity Configure Refer to the description of CCEN_CC1NPOL.
12	CC4EN	R/W	Capture/Compare Channel 4 Output Enable Refer to CCEN_CC1EN
13	CC4POL	R/W	Capture/Compare Channel 4 Output Polarity Configure Refer to the description of CCEN_CC1POL.
14	Reserved		
15	CC4NPOL	R/W	Capture/Compare Channel 4 Output Polarity Configure Refer to the description of CCEN_CC1NPOL.

Table 51 Output Control Bit of Standard Ocx Channel

CcxEN bit	Ocx output state
0	Disable output ((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.

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 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 value currently loaded in the capture/compare register Compare the value CC1 of the capture and 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 comparison results; If the output compare preload is enabled (OC1PEN=1 for TMRx_CCM1 register), the written value will affect the output comparison result when an update event is generated.

15.6.14 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

15.6.15 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

15.6.16 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

15.6.17 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 transmission length and transmission times of DMA in continuous mode. The data transmitted 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=TMR4_CTRL1 (slave address) means the address of the data to be transmitted, while the TMRx_CTRL1 address +DBADDR+7 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: When the transmission data is set to 16 bits, the data will be transmitted to seven registers 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		

15.6.18 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 to 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.

15.6.19 Input Channel Remapping Register (TMR3_OPT) (only for TMR3)

Offset address: 0x50

Reset value: 0x0000

Field	Name	R/W	Description
15:0	RMPSEL	R/W	Timer Input Channel 2 remapping selection: 00: TMR3 Channel 2 connected to GPIO 01: TMR3 Channel 2 connected to RTCCLK 10: TMR3 Channel 2 connected to HSECLK/32 11: TMR3 Channel 2 connected to the Main Clock Output (MCO). This selection is configured by the MCOSEL bit in the Clock Configuration Register RCM_CFG1.

16 Basic Timer (TMR6/7/8)

16.1 Introduction

The basic timer TMR6/7/8 consists of an unsigned 16-bit counter, auto reload register, prescaler and trigger controller.

The basic timer provides a time base for the general-purpose timers. Through configuration, they can generate DMA requests (TMR6/7). Note that TMR8 does not support DMA requests.

16.2 Main Characteristics

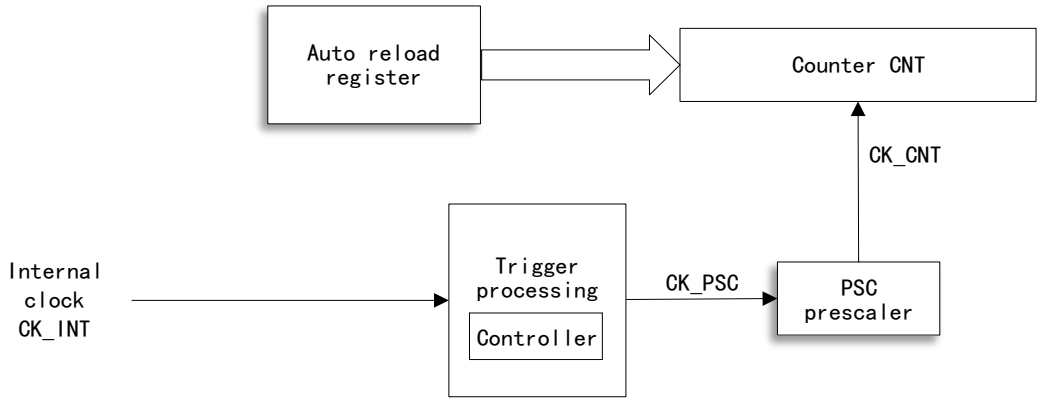
(1) Counter: 16-bit counter, which can only count up

(2) Prescaler: 16-bit programmable prescaler

(3) Clock source: There are only internal clocks

16.3 Structure Block Diagram

Figure 58 Basic Timer Structure Block Diagram



16.4 Functional Description

16.4.1 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.

16.4.2 Timebase unit

The time base unit in the basic timer contains three registers:

- Counter register (CNT) 16 bits

- Autoreload register (AUTORLD) 16 bits
- Prescaler (PSC) 16 bits

Counter CNT

The basic timer only has one count mode: count-up

Count-up mode

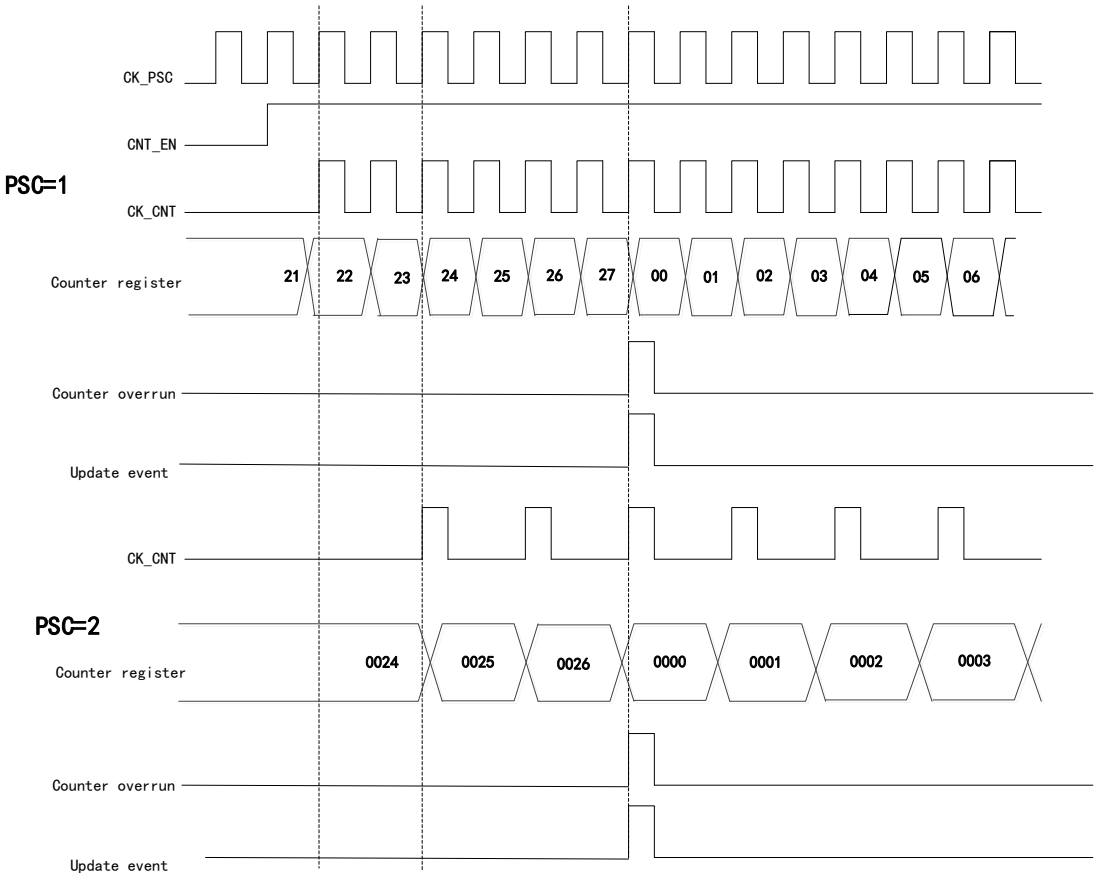
When the counter is in count-up mode, the counter will count up from 0; every time a pulse is generated, the counter will increase by 1 and when the value of the counter (TMRx_CNT) is equal to the value of the auto reload (TMRx_AUTORLD), then the counter will start to count again from 0, a count-up overrun 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 59 Timer Timing Diagram (Internal Clock Division Factor of 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 within 1~65536 (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.

16.5 Register Address Mapping

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

Table 52 TMR6, TMR7 and TMR8 Register Address Mapping

Register name	Description	Offset address
TMRx_CTRL1	Control register 1	0x00
TMRx_DIEN	DMA/Interrupt enable register	0x0C
TMRx_STS	Status register	0x10
TMRx_CEG	Control event generation register	0x14
TMRx_CNT	Counter register	0x24
TMRx_PSC	Prescale register	0x28
TMRx_AUTORLD	Auto reload register	0x2C

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; when it is configured as the trigger mode, it can write 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: Enable update event (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: Disable update event

Field	Name	R/W	Description
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 by this bit. 0: The counter overruns or underruns Set UEG bit Update generated by slave mode controller 1: The counter overruns or underruns Note: TMR8 does not support DMA functionality.
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 subsequent output level of the channel will no long be changed. 0: Disable 1: Enable
6:4	Reserved		
7	ARPEN	R/W	Auto-reload Preload Enable When the buffer is disabled, modification of TMRx_AUTORLD by program will immediately lead to modification of the values loaded to the counter; when the buffer is enabled, modification of TMRx_AUTORLD by program will lead to modification of the values loaded to the counter at the next update event. 0: Disable 1: Enable
15:8	Reserved		

16.6.2 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 Note: TMR8 does not support DMA functionality.
15:9	Reserved		

16.6.3 Status register (TMRx_STS)

Offset address: 0x10

Reset value: 0x0000

Field	Name	R/W	Description
0	UIFLG	RC_W0	Update Event Interrupt Generate Flag 0: No update event interrupt occurs 1: Update event interrupt occurred When the counter value is reloaded or reinitialized, an update event will be generated. The bit is set to 1 by hardware and cleared to 0 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 an update event, and the counter needs to be initialized by software; (3) URSSEL=0 and UD=0 on TMRx_CTRL1 register, and an update event will be generated when the counter is initialized by trigger event.
15:1	Reserved		

16.6.4 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 an update event This bit is set to 1 by software, and cleared to 0 by hardware. Note: When an update event is generated, the counter of the prescaler will be cleared to 0, 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.

16.6.5 Counter register (TMRx_CNT)

Offset address: 0x24

Reset value: 0x0000

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

16.6.6 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)

16.6.7 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.

17 Independent Watchdog Timer (IWDT)

17.1 Introduction

The independent watchdog is used to monitor system faults caused by software errors.

The independent watchdog will initiate a system reset when the counter decrements to zero. Additionally, if the counter is reloaded with a value greater than the window value before reaching zero, a reset will also be triggered.

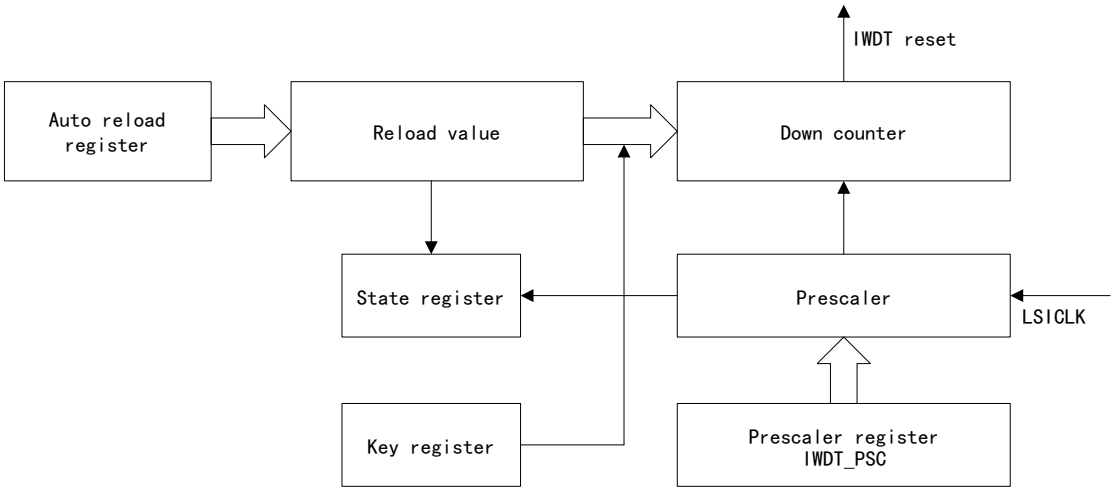
The independent watchdog consists of an 8-bit prescaler IWDT_PSC, a 12-bit down counter, a 12-bit reload register IWDT_CNTRLD, a key register IWDT_KEY, a status register IWDT_STS, and a window register IWDT_WIN.

The independent watchdog has an independent clock source, and even if the master clock fails, it is still valid.

The independent watchdog is applicable when an independent environment is required but the accuracy requirement is not high.

17.2 Functional Block Diagram

Figure 60 Block Diagram of Independent Watchdog



Note: The prescaler, reload value and count-down counter are located in V_{DD} power supply area; the prescaler register, status register, reload register and keyword register are located in 1.2 V power supply area. The watchdog function is in the V_{DD} power supply area and it can work normally in the stop or standby mode.

17.3 Functional Description

17.3.1 Key register

Write 0xCCCC in the key register to enable the independent watchdog, then the counter starts to count down from the reset value 0xFFF 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, reload register and window value register.

17.3.2 Window register

The default value of Window Register IWDT_WIN is 0xFFF. Without update, the window option is disabled. When the window value is changed, the reloading operation will be performed, and the value of watchdog counter will be set to the value of IWDT_CNTRLD, which can delay the cycle needed for reset.

The independent watchdog can work in the window watchdog mode, and the value of window register IWDT_WIN needs to be set appropriately.

17.3.3 Configuration IWDT

Configure IWDT when window register is used

(1) Enable IWDT (write 0xCCCC to the key register IWDT_KEY)

(2) Enable the register access permission (write 0x5555 to the key register IWDT_KEY)

(3) Configure IWDT_PSC prescaler register (write the value within 0~7 to IWDT_PSC)

(4) Wait until the value of status register IWDT_STS is updated to 0x00

(5) Configure the window register IWDT_WIN (the value of auto reload register IWDT_CNTRLD can be updated to the value of watchdog register)

Note: When the value of status register IWDT_STS is 0x00, the window value will be written to refresh the counter by the auto reloaded value.

Configure IWDT when window register is disabled

(1) Enable IWDT (write 0xCCCC to the key register IWDT_KEY)

(2) Enable the register access permission (write 0x5555 to the key register IWDT_KEY)

(3) Configure IWDT_PSC prescaler register (write the value within 0~7 to IWDT_PSC)

(4) Configure reload register IWDT_CNTRLD

(5) Wait until the value of status register IWDT_STS is updated to 0x00

(6) Refresh the watchdog counter using IWDT_CNTRLD register

17.3.4 Register access protection

The prescaler register IWDT_PSC, reload register IWDT_CNTRLD and window register IWDT_WIN have the function of write protection. If you want to rewrite these three 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.

The prescaler register, reload register and window register can be observed through the status register.

17.3.5 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.

17.3.6 Debug mode

The independent watchdog can be configured in debug mode and choose to stop or continue to work. It depends on the IWDT_STS bit of DBGMCU_APB1F register in DBGMCU module.

17.4 IWDT Register Address Mapping

Table 53 IWDT Register Mapping

Register name	Description	Offset address
IWDT_KEY	Key register	0x00
IWDT_PSC	Prescale register	0x04
IWDT_CNTRLD	Counter reload register	0x08
IWDT_STS	Status register	0x0C
IWDT_WIN	Window register	0x10

17.5 IWDT Register Functional Description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

17.5.1 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, IWDT_CNTRLD and IWDT_WIN 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 to enable the watchdog (the hardware watchdog is unrestricted by this command word). The read-out value is 0x0000.
31:16	Reserved		

17.5.2 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 to the IWDT_KEY register, it is allowed to access the register; in the process of writing to this register, only when PSCUFLG=0 for IWDT_STS register, 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		

17.5.3 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 to IWDT_KEY register; in the process of writing this register, this register can be modified only when CNTUFLG=0. In the process of reading this register, only when CNTUFLG=0 in IWDT_STS register, can the read value be valid. The watchdog timeout cycle can be calculated by the reload value and clock prescaler value.
31:12	Reserved		

17.5.4 Status 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 Value Update Flag When the prescaler factor is updated, set 1 by hardware; after the prescaler factor is updated, clear 0 by hardware; the prescaler factor is updated only when the PSCUFLG bit is cleared to 0.
1	CNTUFLG	R	Watchdog Counter Reload Value Update Flag When the counter reload value is updated, set 1 by hardware; after the counter reload value is updated, clear 0 by hardware; the counter reload value is updated only when the CNTUFLG bit is cleared to 0.
2	WINUFLG	R	Watchdog Counter Window Value Update Flag When the window value is updated, this bit is set to 1 by hardware. After the window value of the counter is updated, this bit is cleared by hardware; the window value is valid only when the IWDT_WIN register is enabled.
31:3	Reserved		

17.5.5 Window register (IWDT_WIN)

Offset address: 0x10

Reset value: 0x0000 0FFF (reset in standby mode)

Field	Name	R/W	Description
11:0	WIN	R/W	Set Watchdog Counter Window Value These bits include the window value and the initial value of down counter These bits can be modified only when STS_WINUFLG=0 Reloading the counter between the counter value and the window value can prevent resetting Note: When reading this register, return the value of V_{DD} power supply domain, so if you want to read data, you should ensure STS_WINUFLG=0.
31:12	Reserved		

Note: When the reload setting, prescaler setting and window value resetting are running, if you want to change the reload value, prescaler value and window value, you need to confirm that the relevant flag bits are “0”. There is no need to wait after update, unless you want to enter the low-power mode.

18 Universal Synchronous/Asynchronous Transceiver (USART)

18.1 Full Name and Abbreviation Description of Terms

Table 54 Full Name and Abbreviation Description of Terms

Full Name	Abbreviation
Clear to Send	CTS
Request to Send	RTS
Most Significant Bit	MSB
Least Significant Bit	LSB
Guard	GRD
Overrun	OVR

18.2 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 rates and supports multiprocessor communication.

USART not only supports the standard asynchronous transceiver mode but also supports synchronous one-way communication and some other serial data exchange modes, such as LIN protocol and hardware flow control modes.

USART also supports DMA function to realize high-speed data communication.

18.3 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) Select speed and clock tolerance with programmable 8-time or 16-time oversampling rate

(7) Programmable high or low priority

(8) Independent transmitter and receiver enable bit

(9) Independent signal polarity control transmitter and receiver

(10) Can switch TX/RX pins

(11) Support timeout detection

(12) Programmable baud rate generator, with the baud rate up to 8 Mbit/s

(13) Automatic baud rate detection

(14) Multiprocessor communication:
- If the address does not match, enter the mute mode
- Wake up from mute mode through idle bus detection or address flag detection

(15) Double-clock drive
- Function of wake-up from the stop mode
- Baud rate selection independent of PCLK

(16) Synchronous transmission mode

(17) Generation and detection of LIN break frame

(18) Support hardware flow control and RS485 drive enable

(19) DMA can be used for continuous communication

(20) Status 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

(21) Multiple interrupt sources:
- The transmit register is empty
- Transmission completed
- CTS changed
- The receive register is not empty
- Overrun error
- Bus idle
- Parity check error
- LIN break detection
- Noise error

- Overrun error
- Frame error
- Address/Character match
- Wake up from the stop mode
- Failed to receive interrupt on time

18.4 Functional Description

Table 55 USART Pin Description

Pin	Type	Description
USART_RX	Input	Data receiving
USART_TX	Output I/O (single-line 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
USART_DE	Input	Drive enable activating external transmitter/receiver

18.4.1 Single-line half-duplex communication

The 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 bits of USART_CTRL2 register, and SCEN 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.
- Transmitting data and receiving data can not be carried out at the same time. The data cannot be received before they are transmitted. To receive data, the reception function can be enabled only after TXCFLG bit of USART_STS register is set to 1.
- If there is data collision on the bus, software is required to manage the distributed communication process.

18.4.2 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
- The PCEN bit controls whether to enable the parity check bit

- The PCFG bit controls the parity check bit to determine if it is odd or even

Table 56 USART 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+parity check bit+stop bit
1	0	Start bit+9-bit data+stop bit
1	1	Start bit+8-bit data+ parity check bit+stop bit

Configurable stop bit

Four different stop bits can be configured through the STOPCFG bit of USART_CTRL2 register.

- 1 stop bit: The default stop bit
- 0.5 stop bit
- Two stop bits: Used in normal mode, single-line mode and hardware flow control mode
- 1.5 stop bits

Parity check bit

The 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 parity check: When the number of frame data and check bit “1” is even, the even check bit is “0”; otherwise it is “1”.
- Odd parity check: When the number of frame data and check bit “1” is even, the odd check bit is “1”; otherwise it is “0”.
- Check generation: When transmitting data, set the PCEN bit of USART_CTRL1 register, and the check bit will replace the MSB bit of the data and be transmitted.
- Parity check:
  - If the parity check fails, the PEFLG flag bit of USART_STS register will be set.
  - If the check control is enabled, corresponding interrupt will be triggered. Write 1 to the PECLR bit of USART_INTFCLR register, and PEFLG flag bit can be cleared.

18.4.3 Transmitter

When the 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.

18.4.3.1 Character transmission

During transmission period of USART, the least significant bit (LSB) of the data will be moved out by TX pin first. In this mode, USART_TXDATA register has a buffer between the internal bus and the transmit 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 bit whose number is configurable.

Transmission configuration steps

(1) Decide the word length by setting the DBLCFG bit of USART_CTRL1 register

(2) Decide the number of stop bits by setting the STOPCFG bit of USART_CTRL2 register

(3) If multi-buffer communication is selected, DMA should be enabled in USART_CTRL3 register

(4) Set the baud rate of communication in USART_BR register

(5) Set the UEN bit of USART_CTRL1 register to enable USART. Wait for TXBEFLG bit of USART_STS register to be set to 1

(6) Enable the TXEN bit in USART_CTRL1 register, and transmit an idle frame

(7) Write data to USART_TXDATA register (if DMA is disabled, repeat Step 7 for each byte to be transmitted)

(8) 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 may be destroyed, which is because if the baud rate generator stops counting, the data being transmitted will be lost.

18.4.3.2 Single-byte communication

TXBEFLG bit can be cleared by writing to USART_TXDATA 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 overwriting 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 TXDATA 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 to the USART_TXDATA register, before entering the low-power mode or before disabling the USART module, wait to set TXCFLG to 1.

18.4.3.3 Break frame

It is regarded that the break frames all receive “0” within one frame period. One break frame can be transmitted by setting the TXBFQ bit of USART_REQUEST register, and the length of the break frame is determined by the DBLCFG bit of USART_CTRL1 register. If the TXBFQ bit is set, after completion of transmission of current data, the TX line will send a break frame, and after completion of transmission of break frame, the TXBFQ 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 TXBFQ bit is reset before transmission of the break frame, the break frame will not be transmitted. To transmit two consecutive break frames, the TXBFQ bit should be set after the stop bit of the previous break symbol.

18.4.3.4 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 transmitted before the first data frame.

18.4.4 Receiver

18.4.4.1 Character receiving

During receiving period of USART, RX pin will first introduce the least significant bit (LSB) of the data. In this mode, USART_RXDATA register has a buffer between the internal bus and the receive shift register. The data is transmitted to the buffer bit by bit. When fully receiving the data, the corresponding receive register is not empty, then the user can read USART_RXDATA.

Receiving configuration steps

(1) Program the oversampling rate to 8 or 16 times

(2) Decide the word length by setting the DBLCFG bit of USART_CTRL1 register

(3) Decide the number of stop bits by setting the STOPCFG bit of USART_CTRL2 register

(4) If multi-buffer communication is selected, DMA should be enabled in USART_CTRL3 register

(5) Set the baud rate of communication in USART_BR register

(6) Set the UEN bit of USART_CTRL1 register to enable USART

(7) Set the RXEN bit of USART_CTRL1 to enable reception

Note:

(1) RXEN bit cannot be reset during data receiving period; otherwise, the bytes being received will be lost.

(2) In the process of the receiver receiving a data frame, if an 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_RXDATA 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_RXDATA register by software or by writing 0 to it.

(6) In multi-buffer mode, after each byte is received, the RXBNEFLG bit of USART_STS register will be set to 1, and can be cleared by reading the data register by DMA.

18.4.4.2 Break frame

When the receiver receives a break frame, USART will process it like receiving a frame error.

18.4.4.3 Idle frame

When the receiver receives an idle frame, USART will process it like receiving an ordinary data frame; if the IDLEIEN bit of USART_CTRL1 is set, an interrupt will be generated.

18.4.4.4 Select the clock source

The clock source must be selected by the clock control system before USART is enabled:

(1) The clock source shall be selected according to the transmission speed and the possibility of use of USART in low-power mode.

(2) The clock source frequency is f_{CK}.

- The range of communication speed is determined by the clock source. USART should be enabled before the clock source is selected.
- When USART adopts dual-clock domain or wakes up the stop mode, PCLK, HSICLK or SYSCLK can be used as the clock source; otherwise, the clock source is PCLK.
- If LSICLK is selected as the clock source, USART can receive data even in low-power mode. It can select according to the received data and wake-up mode, and wake up MCU when necessary, so that DMA can read the received data.
- The receiver realizes the data recovery of different oversampling technologies configured by users to distinguish valid incoming data and noises, which requires a trade-off between the maximum communication speed and noise/clock inaccuracy immunity.

18.4.4.5 Oversampling Ratio

The OSMCFG bit of USART_CTRL1 register determines the oversampling rate.

If the oversampling rate is 8 times the baud rate, the speed is high, but the clock tolerance is small. If it is 16 times, the speed is low, but the clock tolerance is big.

18.4.4.6 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 RXDATA register. After the byte is received, the RXBNEFLG bit will be set to 1. This bit needs to be reset before receiving the next data or servicing the previous DMA request; otherwise, an overrun error will occur.

When an overrun error occurs

- The OVREFLG bit of USART_STS is set to 1
- The data in RXDATA register will not be lost
- The data in the shift register previously received will be overwritten, but the data received later will not be saved
- If RXBNEIEN bit or ERRIEN bit of USART_CTRL1 is set, an interrupt will be generated
- When OVREFLG bit is set, it means the data has been lost. There are two possibilities:
  - When RXBNEFLG=1, the previous valid data is still on RXDATA register, and can be read
  - When RXBNEFLG=0, there is no valid data in RXDATA register
- The OVREFLG bit can be reset by reading USART_STS and USART_RXDATA registers.

18.4.4.7 Noise error

When noise is detected in the receiving process of the receiver:

- Set NEFLG flag on the rising edge of RXBNEFLG bit of USART_STS register
- Invalid data is transmitted from the shift register to USART_RXDATA register.
- In single- byte communication, no interrupt will be generated, but in multi-buffer communication, an interrupt will be generated by setting the ERRIEN bit of USART_CTRL3 register

Note: 8x oversampling ratio cannot be used in LIN mode.

18.4.4.8 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 by the receiver in the receiving process:

- Set the FEFLG bit of USART_STS register.
- Invalid data is transmitted from the shift register to USART_RXDATA register.
- In single- byte communication, no interrupt will be generated, but in multi-buffer communication, an interrupt will be generated by setting the ERRIEN bit of USART_CTRL3 register.

18.4.5 Tolerance of receiver to the change of clock

Only when the total clock system deviation is less than the tolerance of USART receiver, can the USART receiver work normally.

Deviation will occur in any of the following circumstances:

(1) DTRA: Deviation caused by transmitter error

(2) DQUANT: Deviation caused by receiver baud rate quantization

(3) DREC: Change of receiver oscillator

(4) DTCL: Deviation caused by transmission line

18.4.6 Baud rate generator

The baud rate division factor (USARTDIV) is a 16-bit number consisting of 12-bit integer and 4-bit fraction. Its relationship with the system clock:

$$\text{Baud rate}=\text{PCLK}/16*(\text{USARTDIV})$$

The value of USARTDIV should not exceed 0xFFF7; otherwise, data may be corrupted.

USART2/3 uses PCLK1 as its system clock, while USART1 uses PCLK2 as its system clock. USART must be enabled after the system clock is enabled in the

clock control unit.

18.4.7 Automatic baud rate detection

When a character is received, USART can detect and automatically set the value of the USART_BR register. Automatic baud rate detection functions when the communication speed of the system is unknown, the clock source with low precision is used, or the clock deviation is not measured to obtain the correct bit rate. The clock source must be compatible with the expected communication speed.

A non-zero baud rate must be written for initialization; confirm the character content, and then enable automatic baud rate detection. ABRDCFG bit of USART_CTRL2 register can be set to select the character content, and the possible character content is:

(1) For all characters starting with 1, in this case, measure the length of the start bit (the duration from the falling edge to the rising edge).

(2) For all characters starting with 10xx, in this case, measure the length of the start bit and the first data bit, the duration of the falling edge, to ensure better accuracy when the signal slew rate is small.

(3) One 0x7F character frame (it can be 0x7F in the first mode of LSB, or 0xFE character in the first mode of MSB). In this case, first detect the baud rate of start bit, then take samples of the bits 0 to 6 at the end of Bit 6, and further take samples of the bit rate of the character of Bit 6.

(4) A 0x55 character frame; in this case, first detect the baud rate of the start bit, then detect the baud rate at the end of Bit 0 data, and finally detect the baud rate at the end of Bit 6 data. Take samples of Bit 0, bits 1 to 6 and Bit 6 respectively.

The ABRDEN bit of USART_CTRL2 register determines whether to enable automatic baud rate detection. After the automatic baud rate detection is enabled, wait for the first character on RX line. After detection, the ABRDFLG flag bit of USART_STS register will be set.

Note:

(1) If the line noise is too loud, correct baud rate cannot be guaranteed. In this case, the BR value may be damaged and the ABRDEFLG flag bit will be set, but FEFLG flag bit will not be set. This situation can also happen if the communication speed and automatic baud rate detection are not compatible.

(2) RXBNEFLG interrupt will be generated after detection.

(3) At any time, automatic baud rate detection may be restarted by resetting the ABRDFLG flag (writing a 0).

(4) USART cannot be disabled during automatic baud rate detection; otherwise, the BR value may be damaged.

(5) If the USART auto-baud rate detection fails, the ABRDEFLG flag is set, but FEFLG in the USART_STS register will not be set.

18.4.8 Multiprocessor communication

In multiprocessor communication, multiple USART are connected to form one 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 to reduce the burden of USART. In mute mode, the LINMEN bit of USART_CTRL2 register, and the SCEN and HDEN bits of USART_CTRL3 register are cleared, any receiving state bit will not be set, and all reception interrupts will be disabled.

When mute mode is enabled, there are two ways to exit the mute mode：

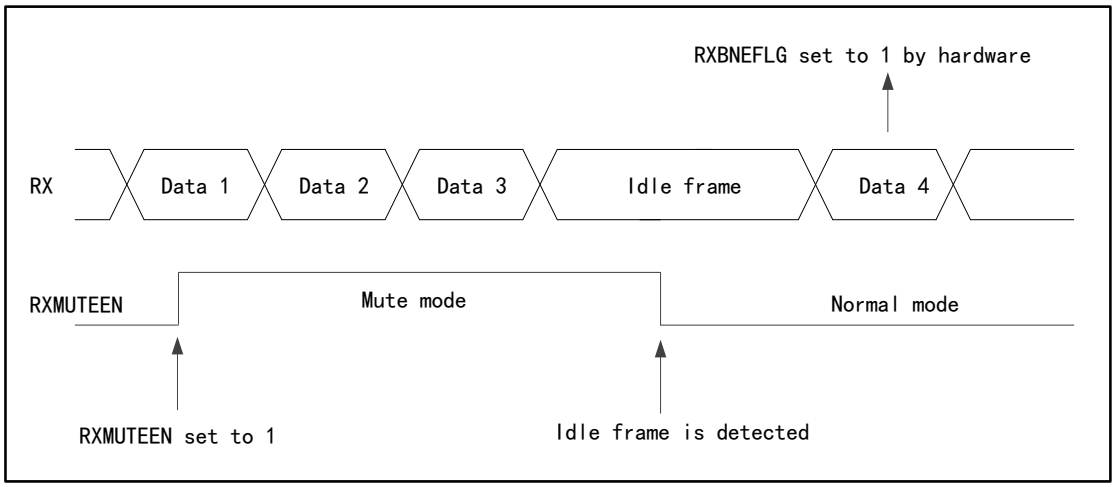
(1) Exit the mute mode when WUPMCFG bit is cleared and the bus is idle.

(2) Exit the mute mode when WUPMCFG bit is set and the address flag is received.

Idle bus detection (WUPMCFG=0)

When RXWFMUTE is set to 1, USART enters the mute mode, and it can wake up from the mute mode when an idle frame is detected, meanwhile, the RXWFMUTE bit will be cleared by hardware. RXWFMUTE can also be cleared through software.

Figure 61 Idle Bus Exits Mute Mode

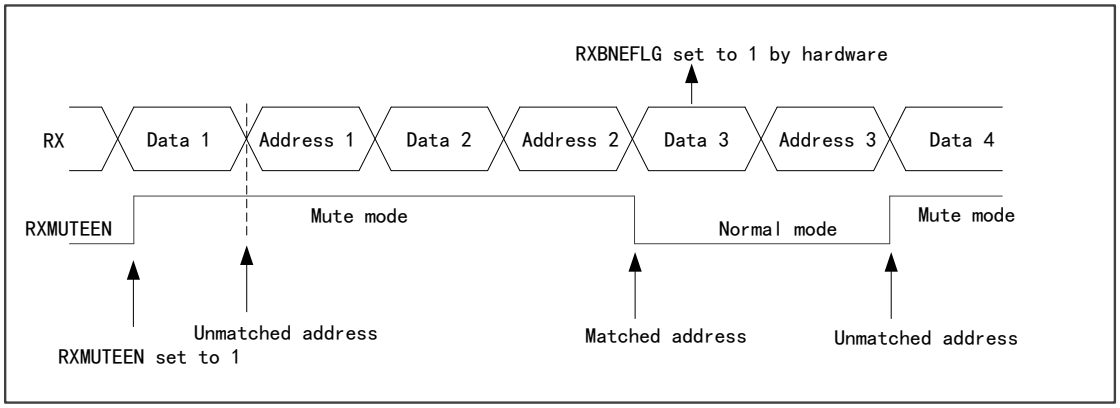


Address flag detection (WUPMCFG=1)

If the address flag bit is 1, this byte is regarded as the address. The lower 4 bits of the address byte store the address. When the receiver receives the address

byte, it will be compared with its own address. 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 62 Address Flag Exits Mute Mode



18.4.9 Wake up from stop mode

When USART uses HSICLK as clock source, USWMEN bit of USART_CTRL1 register decides whether to wake up from the stop mode. Before entering the stop mode, set USWMEN bit of USART_CTRL1 register, and when wake-up event is detected, set WSMFLG to 1, and at this time, an interrupt will be generated as long as WSMIEN is set.

Mute mode in stop mode

It is not allowed to exit from the mute mode during idle detection. If the system exits the mute mode by using the address matching, only the address matching event can be taken as its wake-up source. If the start bit is set to detect wake-up, WSMFLG will be set and RXBNEFLG flag bit will not.

18.4.10 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.

The 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 the HDEN and SCEN bits of USART_CTRL3 register must be cleared.
- The start bit and stop bit of the data frame have no clock output

- Whether the last data bit of the data frame generates USART_CK clock is determined by the LBCPOEN bit of the register USART_CTRL2
- The clock polarity of USART_CK is decided by the CPOL bit of USART_CTRL2 register
- The phase of USART_CK is decided by the CPHA bit of USART_CTRL2 register
- The external CK clock cannot be activated when the bus is idle or a break frame appears

Figure 63 USART Synchronous Transmission Example

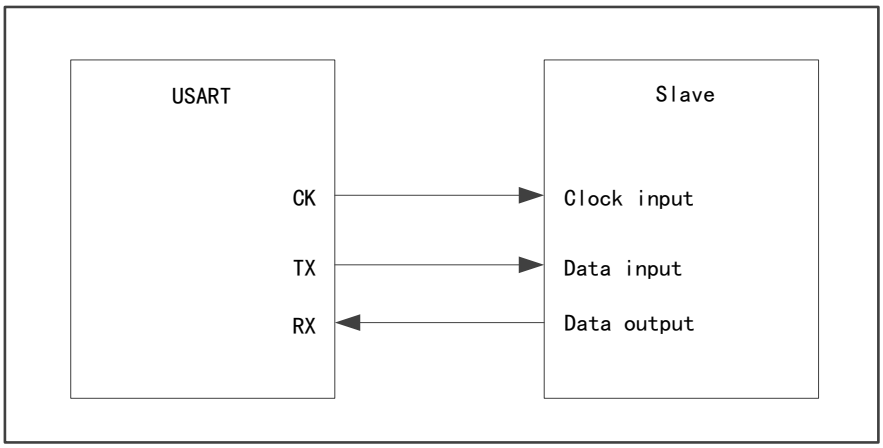


Figure 64 USART Synchronous Transmission Timing Diagram (DBLCFG=10)

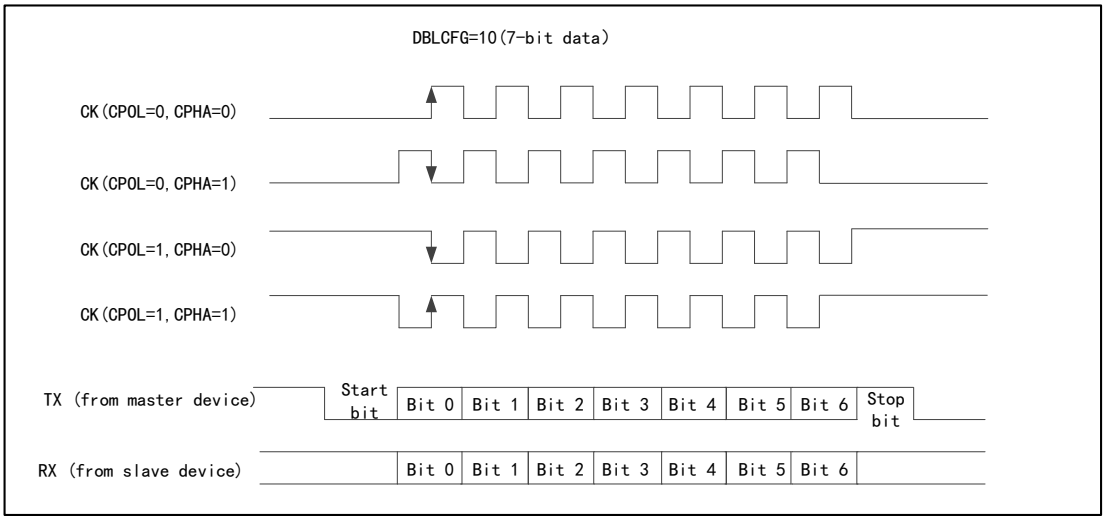


Figure 65 USART Synchronous Transmission Timing Diagram (DBLCFG=00)

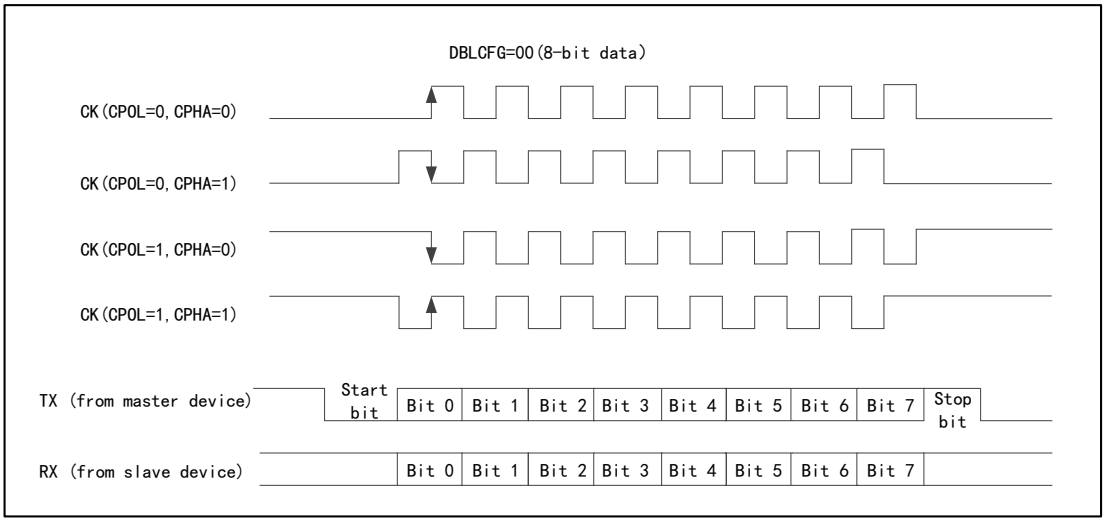
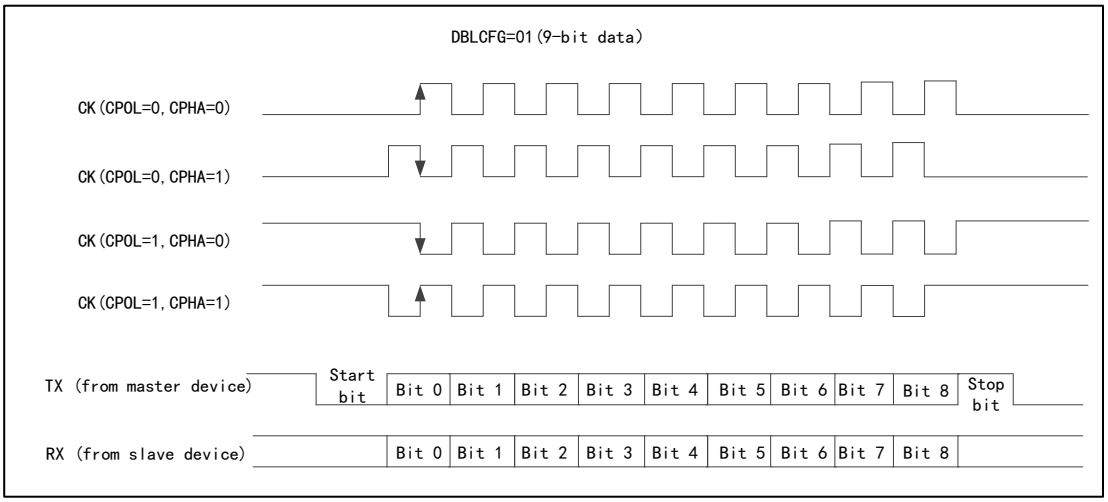


Figure 66 USART Synchronous Transmission Timing Diagram (DBLCFG=01)



18.4.11 LIN mode

The LINMEN bit of USART_CTRL2 register decides whether to enter LIN mode.

When entering LIN mode:

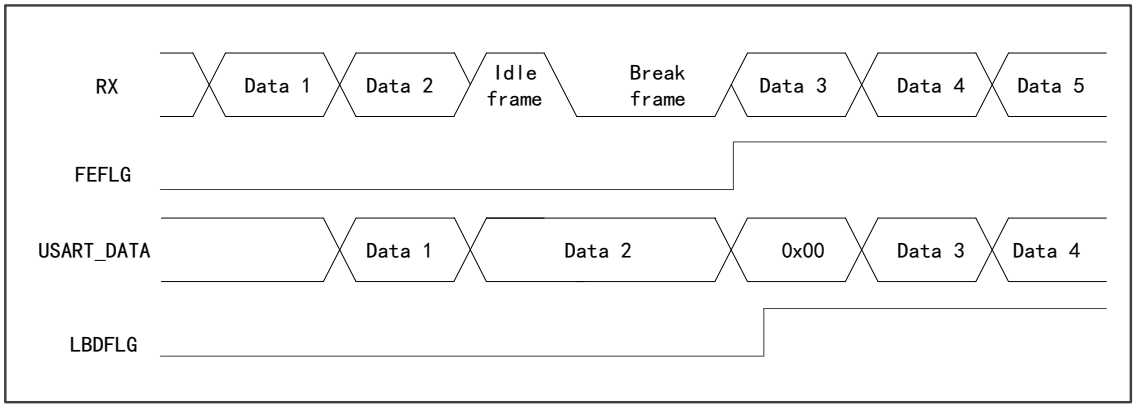
- Each data frame includes 8 data bits and 1 stop bit
- The CLKEN bit and STOPCFG bit of USART_CTRL2 register and the HDEN and SCEN bits of USART_CTRL3 register need to be cleared.

In LIN master mode, USART can generate break frame, and the detection length of break frame can be set to 10 or 11 bits through LBDLCFG bit of USART_CTRL2. The break frame detection circuit is independent of USART receiver, and whether in idle state or in data transmission state, RX pin can detect the break frame, and the LBDFLG bit of USART_STS register is set to 1; at this time, if the 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 error.

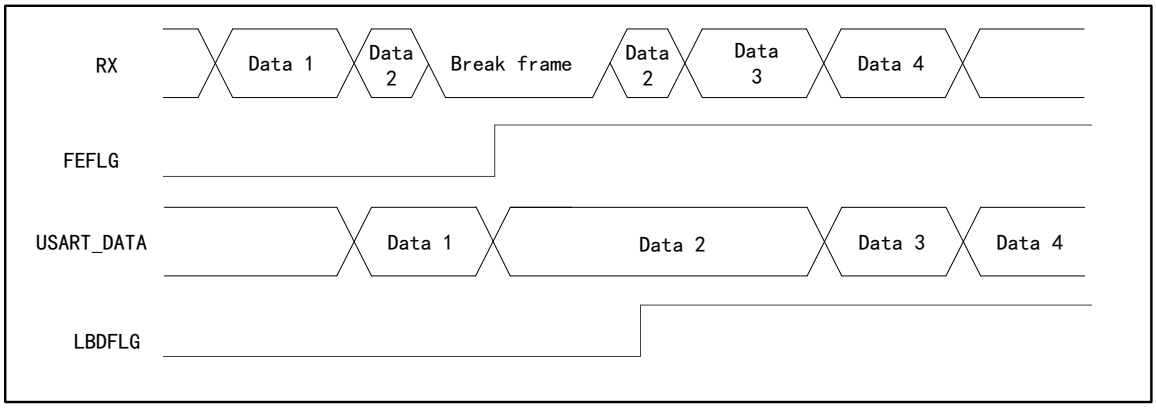
Figure 67 Break Frame Detection in Idle State



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 error.

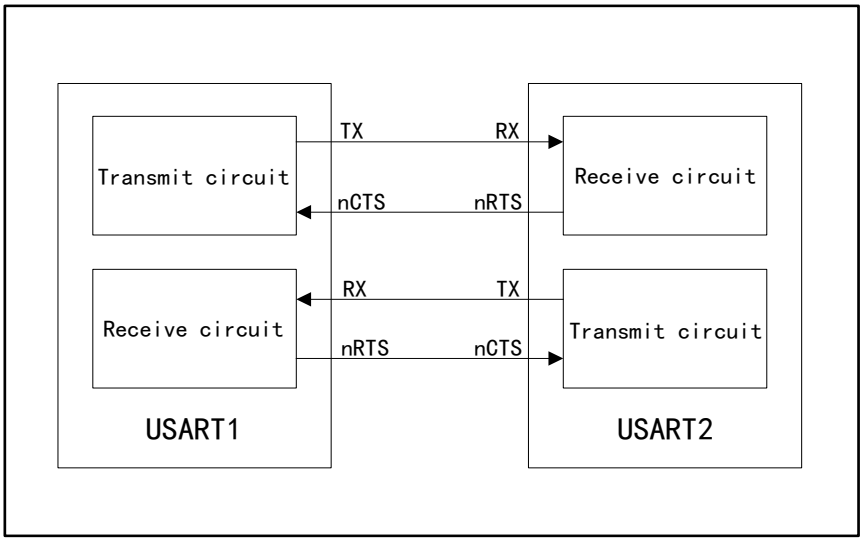
Figure 68 Break Frame Detection in Data Transmission State



18.4.12 Hardware flow control and RS485 drive enable

The function of hardware flow control is to control the serial data stream between two devices through nCTS pin and nRTS pin.

Figure 69 Hardware Flow Control between Two USARTs



CTS flow control

The CTSEN bit of USART_CTRL3 register determines whether to enable CTS flow control. 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, 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

The RTSEN bit of USART_CTRL3 register determines whether to enable RTS flow control. If RTS flow control is enabled, when the receiver receives data, nRTS will be pulled to low. When a data frame is received, nRTS will become high to inform the transmitter to stop transmitting data frame.

RS485 drive enable

The DEN bit of USART_CTRL3 register determines whether to enable the driver enable function, and this function allows DE signal to enable the control terminal of the external transceiver.

Lead time: The time interval between the drive enable signal and the start bit of the first byte. It is controlled by DLTEN[4:0] of USART_CTRL1 register.

Lag time: The time interval between the stop bit of the last byte and the release DE signal. It is controlled by DDLTEN[4:0] of USART_CTRL1 register.

When RS485 driver enable function is activated, the RTS pin will continuously output a low level if the UEN bit in the USART_CTRL1 register is 0.

18.4.13 DMA multi-buffer communication

To reduce the burden of processors, USART can access the data buffer in DMA mode.

Transmission in DMA mode

The 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:

(1) Clear the TXCFLG flag bit of USART_STS register to 0

(2) Set the address of SRAM memory storing data as DMA source address

(3) Set the address of USART_TXDATA register as DMA destination address

(4) Set the number of data bytes to be transmitted

(5) Set channel priority

(6) Set interrupt enable

(7) Enable DMA channel

(8) Wait for TXCFLG bit of USART_STS register to be set to 1, indicating transmission completion

Receive by DMA

The 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:

(1) Set the address of USART_RXDATA register as DMA source address

(2) Set the address of SRAM memory storing data as DMA destination address

(3) Set the number of data bytes to be transmitted

(4) Set channel priority

(5) Set interrupt enable

(6) Enable DMA channel

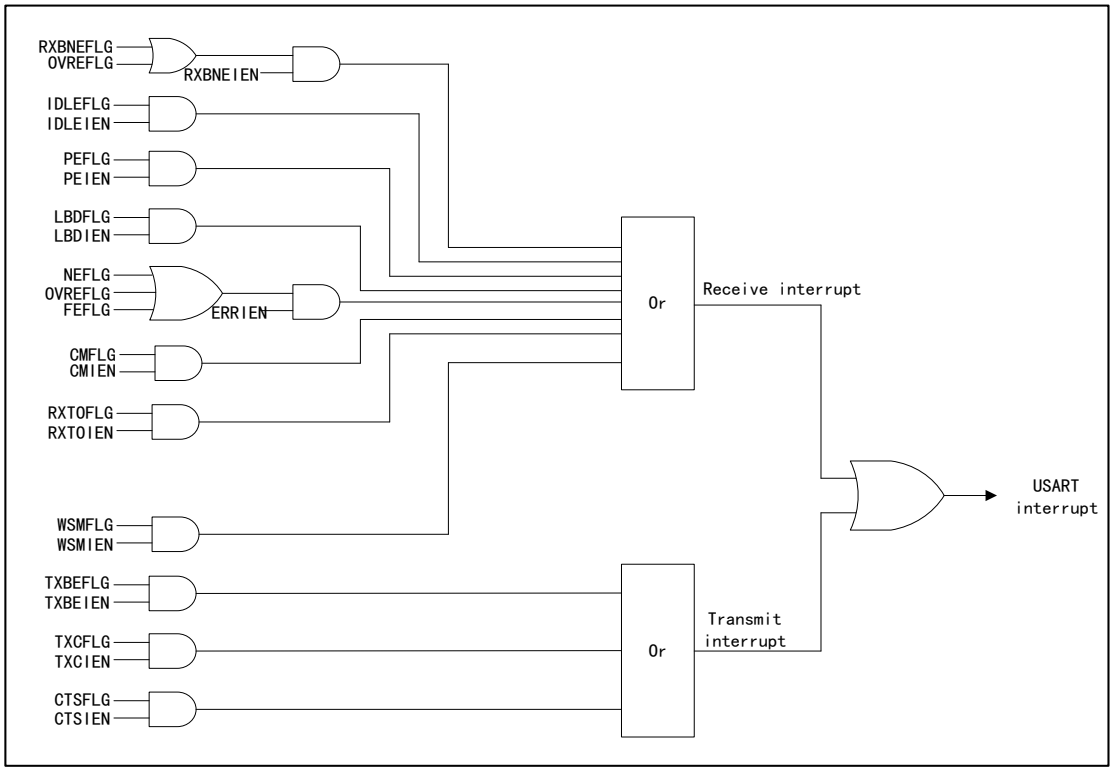
18.4.14 Interrupt request

Table 57 USART Interrupt Request

Interrupt event		Event flag bit	Enable bit
The receive register is not empty		RXBNEFLG	RXBNEIEN
Overrun error		OVREFLG	
Idle line is detected		IDLEFLG	IDLEIEN
Parity check error		PEFLG	PEIEN
LIN break error		LBDFLG	LBDIEN
Receiving error in DMA mode	Noise error	NEFLG	ERRIEN
	Overrun error	OVREFLG	
	Frame error	FEFLG	
Matching character		CMFLG	CMIEN
Error of failing to receive on time		RXTOFLG	RXTOIEN
Stop mode	Wake up from stop mode	WSMFLG	WSMIEN
Data transmit register is empty		TXBEFLG	TXBEIEN
Transmission 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 relation before they are transmitted to the interrupt controller.

Figure 70 USART Interrupt Mapping



18.5 Register Address Mapping

Table 58 USART Register Address Mapping

Register name	Description	Offset address
USART_CTRL1	Control register 1	0x00
USART_CTRL2	Control register 2	0x04
USART_CTRL3	Control register 3	0x08
USART_BR	Baud rate register	0x0C
USART_RXTO	Receive timeout register	0x14
USART_REQUEST	Request register	0x18
USART_STS	Interrupt and status register	0x1C
USART_INTFCLR	Interrupt flag clear register	0x20
USART_RXDATA	Data receive register	0x24
USART_TXDATA	Data transmit register	0x28

18.6 Register Functional Description

18.6.1 Control Register 1 (USART_CTRL1)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	UEN	R/W	Enable USART 0: Disable USART prescaler and output 1: Enable USART module It is set to “1” or cleared by software; clearing this bit will cancel the current operation and the prescaler and output of USART will stop working immediately. The setting of USART will not be reset, but the status flag in USART_STS will be reset.
1	USWMEN	R/W	Enable USART in Stop Mode to Wake Up MCU 0: Disable 1: Enable This bit can be set to 1 and cleared by software; to set this bit, it is required to select HSICLK as the clock source of USART (see the chapter of RCM)
2	RXEN	R/W	Receive Enable 0: Disable 1: Enable, and start to detect the start bit on RX pin Set to 1 or cleared by software.
3	TXEN	R/W	Transmit Enable 0: Disable 1: Enable Set to 1 or cleared by software.
4	IDLEIEN	R/W	IDLE Interrupt Enable 0: Disable 1: Generate an interrupt when IDLEFLG is set Set to 1 or cleared by software.
5	RXBNEIEN	R/W	Receive Buffer Not Empty Interrupt Enable 0: Disable 1: Generate an interrupt when OVREFLG or RXBNEFLG is set Set to 1 or cleared by software.
6	TXCIEN	R/W	Transmit Complete Interrupt Enable 0: Disable 1: Generate an interrupt when TXCFLG is set Set to 1 or cleared by software.
7	TXBEIEN	R/W	Transmit Buffer Empty Interrupt Enable 0: Disable 1: Generate an interrupt when TXBEFLG is set Set to 1 or cleared by software.
8	PEIEN	R/W	Parity Error interrupt Enable 0: Disable 1: Generate an interrupt when PEFLG is set Set to 1 or cleared by software.

Field	Name	R/W	Description
9	PCFG	R/W	Odd/Even Parity Configure 0: Even parity check 1: Odd parity check Set to 1 or cleared by software. The selection will not take effect until the current transmission of bytes is completed. This bit can be set only when USART is not enabled.
10	PCEN	R/W	Parity Control Enable 0: Disable 1: Enable If this bit is set, a parity check bit will be inserted in the most significant bit (MSB) 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. This bit can be set only when USART is not enabled.
11	WUPMCFG	R/W	Wakeup Method Configure 0: Idle bus wakeup 1: Address tag wakeup Set to 1 or cleared by software. This bit can be set only when USART is not enabled.
12	DBLCFG0	R/W	Data Bits Length Configure Set to 1 or cleared by software. This bit can be set only when USART is not enabled.
13	RXMUTEEN	R/W	Receive Mute Mode Enable 0: Normal operating mode 1: Able to switch between normal mode and mute mode Set to 1 or cleared by software.
14	CMIEN	R/W	Character Match Interrupt Enable 0: Disable 1: Generate an interrupt when CMFLG is set Set to 1 or cleared by software.
15	OSMCFG	R/W	Oversampling Mode Configure 0: 16-time oversampling 1: 8-time oversampling This bit can be set only when USART is not enabled.
20:16	DDLTEN[4:0]	R/W	Driver De-lead Time Enable This field is the time interval between the last stop bit and DE signal during transmission. Its unit is sampling time, determined by oversampling rate. If write operation is performed for USART_TXDATA within DDLTEN time, the just written data will be transmitted only after DDLTEN and DLTEN time. This field can be set only when USART is not enabled.
25:21	DLTEN[4:0]	R/W	Driver Lead Time EnableThis field is the time interval between DE signal and the first start bit during transmission. Its unit is sampling time, determined by oversampling rate. This field can be set only when USART is not enabled.

Field	Name	R/W	Description
26	RXTOIEN	R/W	Receiver Timeout Interrupt Enable 0: Disable 1: Generate an interrupt when RXTOFLG is set Set or cleared to 0 by software.
31:27	Reserved		

18.6.2 Control Register 2 (USART_CTRL2)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	Reserved		
4	ADDRLEN	R/W	Slave Address Length Configure 0:4-bit address 1: 7-bit address This field can be set only when USART is not enabled.
5	LBDLCFG	R/W	LIN Break Detection Length Configure 0: 10 bits 1: 11 bits This bit can be set only when USART is not enabled. Note: The Break length depends on the stop bit length.
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 can be set only when USART is not enabled.
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 can be set only when USART is not enabled.
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 can be set only when USART is not enabled.
11	CLKEN	R/W	Clock Enable (CK pin) 0: Disable 1: Enable

Field	Name	R/W	Description
			This bit can be set only when USART is not enabled.
13:12	STOPCFG	R/W	STOP Bit Configure 00: 1 stop bit 01: 0.5 stop bit 10: 2 stop bits 11: 1.5 stop bits This bit can be set only when USART is not enabled.
14	LINMEN	R/W	LIN Mode Enable 0: Disable 1: Enable Set or cleared to 0 by software. In LIN mode, TXBFQ bit can be set to transmit and detect LIN synchronous break symbol. This bit can be set only when USART is not enabled.
15	SWAPEN	R/W	Swap TX/RX Pins Function Enable 0: Use according to standard allocation 1: The functions of TX and RX pins can be exchanged for use, and they will work when crossing and interconnecting with other USART. Set or cleared to 0 by software. This bit can be set only when USART is not enabled.
16	RXINVEN	R/W	RX Pin Active Level Inversion Enable 0: Standard logic level (V_{DD}=1/IDLE, Gnd=0/mark) 1: (V_{DD}=0/mark, Gnd=1/IDLE), which works when there is an external phase inverter on RX line. Set or cleared to 0 by software. This bit can be set only when USART is not enabled.
17	TXINVEN	R/W	TX Pin Active Level Inversion Enable 0: Standard logic level (VDD=1/IDLE, Gnd=0/mark) 1: (V_{DD}=0/mark, Gnd=1/IDLE), which works when there is an external phase inverter on TX line. Set or cleared to 0 by software. This bit can be set only when USART is not enabled.
18	BINVEN	R/W	Binary Data Inversion Enable 0: Positive/direct logic (0=L,1=H) 1: Negative/reverse logic (0=H, 1=L) Set or cleared to 0 by software. This bit can be set only when USART is not enabled. The parity check bit will be inverted when this bit is set.
19	MSBFEN	R/W	Enable Most Significant Bit First Bit Transmission 0: The data of Bit 0 immediately follows the start Bit 1: The data of the most significant bit (MSL) immediately follows the start bit Set or cleared to 0 by software. This bit can be set only when USART is not enabled.
20	ABRDEN	R/W	Auto Baud Rate Detection Enable 0: Disable

Field	Name	R/W	Description
			1: Enable Set or cleared to 0 by software.
22:21	ABRDCFG	R/W	Auto Baud Rate Detection Mode Configure 00: Measure the start bit 01: Measure the falling bit 10: 0x7F frame detection 11: 0x55 frame detection Set or cleared to 0 by software. Note: The baud rate must be within the specified range.
23	RXTODEN	R/W	Receive Timeout Detection Function Enable 0: Disable 1: Enable Set or cleared to 0 by software. Set this bit, and when it is found that the RX line is idle for the length of time configured by RXTO register, the RXTOFLG bit will be set by hardware.
27:24	ADDRL	R/W	USART Device Node Address Low Setup This field is used for wake-up detection of 7-bit address flag during multi-computer communication when entering the mute state or stop mode. This bit can be set only when the receiver is disabled or USAR is not enabled.
31:28	ADDRH	R/W	USART Device Node Address High Setup This field is not only used for wake-up detection of 7-bit address flag during multi-computer communication when entering the mute state or stop mode. (The most significant bit of the character of the transmitter should be 1) It is also used for character detection in normal receiving process. (Then the mute state is disabled) Then if the received 8-bit byte matches ADDRH, CMFLG bit will be set. This bit can be set only when the receiver is disabled or USART is not enabled.

18.6.3 Control Register 3 (USART_CTRL3)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable; when any bit among FEFLG, OVREFLG and NEFLG is set, an interrupt will be generated.
2:1	Reserved		
3	HDEN	R/W	Half-duplex Mode Enable 0: Disable 1: Enable This bit can be set only when USART is not enabled.
5:4	Reserved		

Field	Name	R/W	Description
6	DMARXEN	R/W	DMA Receive Enable 0: Disable 1: Enable Set or cleared to 0 by software.
7	DMATXEN	R/W	DMA Transmit Enable 0: Disable 1: Enable Set or cleared to 0 by software.
8	RTSEN	R/W	RTS 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. This bit can be set only when USART is not enabled.
9	CTSEN	R/W	CTS Function Enable 0: Disable 1: Enable CTS: Clear To Send, which is input signal Only when CTS input signal is low, the data can be transmitted; otherwise, the data cannot be transmitted; if CTS signal is pulled to high during data transmission, 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 can be set only when USART is not enabled.
10	CTSIEN	R/W	CTS Interrupt Enable 0: Disable 1: Generate an interrupt when CTSFLG is set
11	SAMCFG	R/W	Sample Method Configure 0: Sampling for three times 1: Single sampling; flag of noise detection disabled This bit can be set only when USART is not enabled.
12	OVRDEDIS	R/W	Overrun Detection Disable 0: Enable When RXBNEFLG bit is set and new data is received, OVREFLG bit will be set. 1: Disable. When new data are received, if RXBNEFLG is still set but OVREFLG is not set, the data not read will be overwritten by new data. This bit can be set only when USART is not enabled.

Field	Name	R/W	Description
13	DDISRXEEN	R/W	DMA Disable on Receive Error Enable 0: Not disable DMA. The corresponding error flag bit will be set, but in order to avoid data from overrunning and being overwritten, RXBNEFLG will not be set. 1: Disable DMA. If RXBNEFLG is set, the corresponding error flag bit will also be set. DMA request will be unmasked only when the corresponding error flag bit is cleared. Therefore, it is required to first disable DMA request or first clear RXBNEFLG flag and then clear the error flag. This bit can be set only when USART is not enabled.
14	DEN	R/W	Driver Enable Users are allowed to activate the control terminal of external transceiver through DE signal. 0: Disable DE function 1: Enable DE function, and output DE signal on RTS pin This bit can be set only when USART is not enabled.
15	DPCFG	R/W	Driver Polarity Configure 0: DE signal is active high 1: DE signal is active low This bit can be set only when USART is not enabled.
19:16	Reserved		
21:20	WSIFLGSEL	R/W	Wakeup From Stop Mode Interrupt Flag Select 00: When address matches 01: Reserved 10: When start bit is detected 11: When the receive data register is not empty This bit can be set only when USART is not enabled.
22	WSMIEN	R/W	Enable Wakeup Interrupt from Stop Mode 0: Disable 1: Generate an interrupt when WSMFLG is set Set or cleared to 0 by software.
31:23	Reserved		

18.6.4 Baud rate register (USART_BR)

This register can be set only when USART is not enabled. This bit may be reset by hardware during automatic baud rate detection.

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
3:0	FBR	R/W	Fraction of USART Baud Rate Divider Factor The fractional part of the USART baud rate divider (USARTDIV[3:0]) is determined by these 4 bits.
15:4	IBR	R/W	Integer of USART Baud Rate Divider Factor The integer part of the USART baud rate divider (USARTDIV[15:4]) is determined by these 12 bits.
31:16	Reserved		

Note: The value of this register must not exceed 0xFFF6.

18.6.5 Receive timeout register (USART_RXTO)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
23:0	RXTO	R/W	Receiver Timeout Value Setup This field specifies the receive timeout value in the unit of baud clock. In standard mode, after the last byte is received, if no new start bit is detected within the duration of RXTO value, RXTOFLG will be set by hardware.
31:24	Reserved		

18.6.6 Request register (USART_REQUEST)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ABRDQ	W	Auto Baud Rate Detection Request Set this bit, the ABRDFLG flag will be cleared and an automatic baud rate detection will be conducted when receiving data next time.
1	TXBFQ	W	Transmit Break Frame Request Set this bit, TXBFFLG flag will be set and a break frame will be transmitted after the transmission state machine is enabled.
2	MUTEQ	W	Mute Mode Request Set this bit to enter the mute mode and clear RXWFMUTE flag.
3	RXDFQ	W	Receive Data Flush Request Set this bit and RXBNEFLG flag will be cleared. The data that has not been read out in the receive register can be discarded to avoid overrun error
31:4	Reserved		

18.6.7 Interrupt and status register (USART_STS)

Offset address: 0x1C

Reset value: 0x0000 00C0

Field	Name	R/W	Description
0	PEFLG	R	Parity Error Occur Flag 0: No error 1: Parity error is detected In receiving mode, when a parity error occurs, it is set to 1 by hardware; it can be cleared by setting PECLR.
1	FEFLG	R	Frame Error Occur Flag 0: No frame error 1: Frame error or break symbol is detected When there is synchronous dislocation, too much noise or break symbol, it is set to 1 by hardware; this bit can be cleared by setting FECLR.

Field	Name	R/W	Description
2	NEFLG	R	Noise Error Occur Flag 0 : No noise 1: Noise is detected When there is noise error, this bit is set to 1 by hardware; this bit can be cleared by setting NFCLR.
3	OVREFLG	R	Overrun Error Occur Flag 0: No overrun error 1: Overrun error is detected When the RXBNEFLG bit is set and the data in the shift register is to be transmitted to the receive register, it is set to 1 by hardware; this bit can be cleared by setting OVRECLR.
4	IDLEFLG	R	IDLE Line Detected Flag 0: Idle bus is not detected 1: Idle bus is detected When idle bus is detected, this bit is set to 1 by hardware; this bit can be cleared by setting IDLECLR.
5	RXBNEFLG	R	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 receive shift register, it is set to 1 by hardware; this bit can be cleared by reading the RXDATA register or setting RXDFQ.
6	TXCFLG	R	Transmit Data Complete Flag 0: Transmitting data is not completed 1: Transmitting data is completed After the last frame of data is transmitted and the TXBEFLG is set, it is set to 1 by hardware; this bit can be cleared by writing to TXDATA register or setting TXCCLR.
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 transmit data register, this bit is set to 1 by hardware; this bit can be cleared by writing to TXDATA register.
8	LBDFLG	R	LIN Break Detected Flag 0: LIN break is not detected 1: LIN break is detected When LIN break is detected, this bit is set to 1 by hardware; this bit can be cleared by setting LBDCLR. If LBDIEN in USART_CTRL2 is set, an interrupt will be generated.
9	CTSFLG	R	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, this bit is set to 1 by hardware; this bit can be cleared by setting CTSCLR.
10	CTSCFG	R	CTS Status Configure

Field	Name	R/W	Description
			0: Set nCTS line 1: Reset nCTS line This bit is set to 1 or cleared by hardware. This bit sets the reversed state of nCTS input pin.
11	RXTOFLG	R	Receiver Timeout Flag 0: Not timed out 1: Timed out If the start bit is not detected within the duration set by RXTO bit, this bit is set to 1 by hardware; this bit can be cleared by setting RXTOCLR bit.
13:12	Reserved		
14	ABRDEFLG	R	Auto Baud Rate Detection Error Flag This bit is set to 1 by hardware when baud rate detection fails; this bit can be cleared by setting ABRDQ bit. 0: No automatic baud rate detection error 1: Automatic baud rate detection error
15	ABRDFLG	R	Auto Baud Rate Detection Flag When the automatic baud rate function is enabled or the automatic baud rate operation is interrupted, it is set to 1 by hardware; this bit can be cleared when resuming baud rate detection. 0: No automatic baud rate detection or successful automatic baud rate detection 1: Automatic baud rate detection or automatic baud rate detection failed
16	BSYFLG	R	Busy Flag 0: Idle state 1: In the process of receiving data This bit is set to 1 by hardware when the start bit is detected, and it will be cleared after receiving is over. This bit is set to 1 or cleared by hardware.
17	CMFLG	R	Character Match Flag 0: No character matches 1: There is matching character When the received character matches the value set by ADDR[7:0], this bit is set to 1 by hardware; this bit can be cleared by setting CMCLR bit.
18	TXBFFLG	R	Transmit Break Frame Flag 0: Not transmit 1: Will transmit If TXBFQ bit is set, this bit can be set to 1 by software; when transmitting the stop bit of the break frame, this bit is cleared by hardware.
19	RXWFMUTE	R	Receiver Wakeup from Mute Mode 0: Normal mode 1: Mute mode When switching the wake-up mode and the mute mode, this bit is set to 1 and cleared by hardware; if it is wakened up by idle

Field	Name	R/W	Description
			signal, this bit is set to 1 by writing to USART_REQUEST register. WUPMCFG bit determines the control sequence of mute mode.
20	WSMFLG	R	Wakeup From Stop Mode Flag 0: Not detected 1: Detected This bit can be cleared by setting PECLR bit. If WSMFLG bit is set, an interrupt will be generated.
21	TXENACKFLG	R	Transmit Enable Acknowledge Flag Set to 1 by hardware when reading the transmit enable signal. The idle frame request will be generated when TXEN=0. To ensure minimum cycle of TXEN=0, TXEN will be set immediately.
22	RXENACKFLG	R	Receive Enable Acknowledge Flag Set to 1 by hardware when reading the receive enable signal. This bit is used to confirm whether USART has been ready to receive data before entering the stop mode.
31:23	Reserved		

18.6.8 Interrupt flag clear register (USART_INTFCLR)

Offset address: 0x20

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	PECLR	RC_W1	Parity Error Flag Clear Set this bit and the PEFLG flag bit of USART_STS register can be cleared.
1	FECLR	RC_W1	Framing Error Flag Clear Set this bit and the FEFLG flag bit of USART_STS register can be cleared.
2	NECLR	RC_W1	Noise Detected Flag Clear Set this bit and the NEFLG flag bit of USART_STS register can be cleared.
3	OVRECLR	RC_W1	Overrun Error Flag Clear Set this bit and the OVREFLG flag bit of USART_STS register can be cleared.
4	IDLECLR	RC_W1	IDLE Line Detected Clear Flag Set this bit and the IDLEFLG flag bit of USART_STS register can be cleared.
5	Reserved		
6	TXCCLR	RC_W1	Transmission Data Complete Flag Clear Set this bit and the TXCFLG flag bit of USART_STS register can be cleared.
7	Reserved		
8	LBDCLR	RC_W1	LIN Break Detection Flag Clear

Field	Name	R/W	Description
			The LBDFLG flag bit of USART_STS register can be cleared by setting this bit.
9	CTSCLR	RC_W1	CTS Flag Clear Set this bit and the CTSFLG flag bit of USART_STS register can be cleared.
10	Reserved		
11	RXTOCLR	RC_W1	Receiver Timeout Flag Clear Set this bit and the RXTOFLG flag bit of USART_STS register can be cleared.
16:12	Reserved		
17	CMCLR	RC_W1	Character Match Flag Clear Set this bit and the CMFLG flag bit of USART_STS register can be cleared.
19:18	Reserved		
20	WSMCLR	RC_W1	Wakeup From Stop Mode Flag Clear WSMFLG flag bit of USART_STS register can be cleared by setting this bit.
31:21	Reserved		

18.6.9 Data receive register (USART_RXDATA)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
8:0	RXDATA	R	Receive Data Value Setup Include the received data byte. Provide the parallel interface between input shift register and internal bus. If the parity check bit is turned on when receiving data, read this register and the most significant bit is the check bit.
31:9	Reserved		

18.6.10 Transmit data register (USART_TXDATA)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
8:0	TXDATA	R/W	Transmit Data Value Setup Include the data byte to be transmitted. Provide the parallel interface between transmit shift register and internal bus. If the parity check bit is turned on when transmitting data, it is invalid to write to the most significant bit (MSB), as it will be replaced by the check bit and transmitted.
31:9	Reserved		

19 Serial Peripheral Interface (SPI)

19.1 Full Name and Abbreviation Description of Terms

Table 59 Full Names and Abbreviations of SPI Terms

Full Name	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
Transmit	TX
Receive	RX
Busy	BSY

19.2 Introduction

SPI interface can be configured to support SPI protocol.

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.

19.3 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 the format of 4-bit to 16-bit transmission frame

(4) Support multi-master device mode

(5) Support special transmission and receiving mark and can trigger interrupts

(6) Have SPI bus busy state flag

(7) SPI supports Motorola mode

(8) Fast communication in master/slave mode, up to 8 MHz

(9) Clock polarity and phase are programmable

(10) Data sequence is programmable; select MSB or LSB in front

(11) An interrupt can be triggered by master mode fault or overrun

(12) Have DMA transmit and receive buffers

(13) Calculate, transmit and verify by hardware CRC

(14) CRC error flag

(15) Two 32-bit embedded RXFIFO and TXFIFO have DMA function

19.4 SPI Functional Description

19.4.1 Description of SPI signal line

Table 60 SPI Signal Line Description

Pin name	Description
SCK	Master device: SPI clock output Slave device: SPI clock input
MISO	Master device: Input the pin and receive data Slave device: Output the pin and transmit data Data direction: From slave device to master device
MOSI	Master device: Output the pin and transmit 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. Hardware NSS mode of master device hardware: NSS outputs, in single-master mode, NSS OFF output: Operation of multiple master environments is allowed, Slave hardware NSS mode: The NSS signal is set to low as the chip selection signal of the slave

19.4.2 Communication format

In SPI communication, receiving data and transmitting data can be carried out at the same time. SCK transmits and samples the data on the data line synchronously. The communication format depends on the clock phase, clock polarity and data frame format. If the communication is normal, the master device and the slave device must be in the same communication format.

19.4.2.1 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 low in idle state
- When CPOL=1, SCK signal line is high in idle state

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 61 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
3	1	1	Even edge	High level

Note:

(1) To change CPOL and CPHA bits, SPI must be disabled by clearing the SPIEN bit to 0.

(2) When SCK is in idle state, if CPOL=1, pull up SCK; if CPOL=0, pull down SCK.

19.4.2.2 Data frame format

Select LSB or MSB first by configuring LSBSEL bit of SPI_CTRL1 register. Select the data word length by configuring DSCFG bit of SPI_CTRL2 register; no matter which data word length is selected; it must be aligned with FRTCFG when performing read access to FIFO. When accessing SPI_DATA register, the data frames are always right aligned. In the process of communication, only the bits within the data word length range will be output with the clock.

19.4.3 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:

- Enable NSS output: When SPI is in master mode, enable SSOEN bit, NSS pin will be pulled to low and SPI will automatically enter the slave mode.
- Disable NSS output: Operation is allowed in multi-master environments.

19.4.4 SPI mode

19.4.4.1 Initialization of SPI master mode

In master mode, serial clock is generated on SCK pin.

Configuration of master mode:

- Configure MSMCFG=1 in SPI_CTRL1 register, and set to master mode
- Select the serial clock baud rate by configuring BRSEL bit in SPI_CTRL1 register
- Select the polarity and phase by configuring CPOL and CPHA bits in SPI_CTRL1 register
- Select the transmission mode by configuring RXOMEN, BMOEN and BMEN bits in SPI_CTRL1 register
- Select the data bit width by configuring DSCFG bit in SPI_CTRL2 register
- Enable NSS pulse mode by configuring NSSPEN bit in SPI_CTRL2 register (when configuring this bit, CPHA bit must be set to 1)
- Set RXFIFO threshold value for trigging RXBNEFLG event by configuring FRTCFG bit in SPI_CTRL2 register
- If DMA function is used, it is required to configure LDTX and LDRX bits of SPI_CTRL2 register
- If CRC is used, it is required to set CRC polynomial as input and also set CRCEN bit
- 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
- Configure SPIEN bit in SPI_CTRL1 register to enable SPI

In master mode: MOSI pin is data output, while MISO is data input.

19.4.4.2 Initialization of SPI slave mode

In slave mode, SCK pin receives the serial clock from the master device.

Configuration of slave mode:

- Configure MSMCFG=0 in SPI_CTRL1 register, and set to slave mode
- Select the polarity and phase by configuring CPOL and CPHA bits in SPI_CTRL1 register
- Select the transmission mode by configuring RXOMEN, BMOEN and BMEN bits in SPI_CTRL1 register
- Select the data bit width by configuring DSCFG bit in SPI_CTRL2 register
- Enable NSS pulse mode by configuring NSSPEN bit in SPI_CTRL2 register (when configuring this bit, CPHA bit must be set to 1)
- Set RXFIFO threshold value for trigging RXBNEFLG event by configuring FRTCFG bit in SPI_CTRL2 register
- If DMA function is used, it is required to configure LDTX and LDRX bits of SPI_CTRL2 register
- If CRC is used, it is required to set CRC polynomial as input and also set CRCEN bit
- Select LSB or MSB first by configuring LSBSEL in SPI_CTRL1 register
- NSS configuration:
  - In hardware mode: NSS pin must be low in the whole data frame transmission process
  - In software mode: Set SSEN bit in SPI_CTRL1 register and clear ISSEL bit
- Configure SPIEN bit in SPI_CTRL1 register to enable SPI

In slave mode: MOSI pin is data input, while MISO is data output.

19.4.4.3 Full-duplex communication of SPI

Usually, SPI is configured as full-duplex communication, and the master and the slave shift registers are connected through two unidirectional lines MOSI and MISO. During SPI communication, synchronous data transmission is conducted according to SCK clock edge. The data of the master are transmitted to the slave through MOSI pin, and the data of the slave are transmitted to the master through MISO pin. When the data transmission is completed, it means that the information is exchanged successfully.

19.4.4.4 Half-duplex communication of SPI

One clock line and one bidirectional data line

- Enable this mode by setting BMEN bit 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

19.4.4.5 Simplex communication of SPI

One clock line and one unidirectional data line (receive-only or transmit-only)

In this mode, SPI module is used as receive-only or transmit-only.

Transmit-only mode:

- Data are transmitted on transmitting pin (MOSI in master mode, MISO in slave mode)
- Then the receive pin can be used as general-purpose I/O (MISO in master mode, MOSI in slave mode)

Receive-only mode:

- Disable SPI output function by setting RXOMEN bit in SPI_CTRL1 register
- Release the transmit pin (MOSI in master mode, MISO in slave mode)
- In master mode, enable SPI to start communication, clear SPIEN bit of SPI_CTRL1 register and receiving data can be stopped immediately, not needing to read BSYFLG flag (always 1)
- In slave mode: Pull NSS to low, and as long as SCK is pulsed by clock, SPI will always receive

19.4.4.6 Communication of multiple slave devices of SPI

SPI can be operated by multiple slave devices. The master device uses GPIO pin to manage the chip selection line of the slave device, and can control two or more independent slave devices.

The master device decides using which slave device to transmit data by pulling down the NSS pin of the slave device.

19.4.5 Data transmission and receiving process in different modes of SPI

Table 62 Run Modes of SPI

Mode	Configuration	Data pin
Full-duplex mode of master device	BMEN=0, RXOMEN=0	MOSI transmits; MISO receives
Unidirectional receiving mode of master device	BMEN=0，RXOMEN=1	MOSI is not used; MISO receives
Bidirectional transmitting mode of master device	BMEN=1，BMOEN=1	MOSI transmits; 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; MISO transmits
Unidirectional receiving mode of slave device	BMEN=0，RXOMEN=1	MOSI receives; MISO is not used

Mode	Configuration	Data pin
Bidirectional transmitting mode of slave device	BMEN=1，BMOEN=1	MOSI is not used; MISO transmits
Bidirectional receiving mode of slave device	BMEN=1，BMOEN=0	MOSI receives; MISO is not used

Figure 71 Connection in Full Duplex Mode

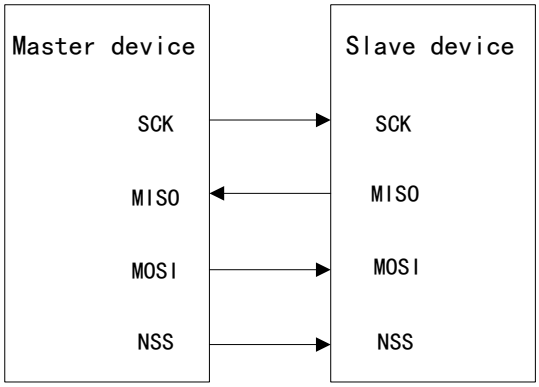


Figure 72 Connection in Half-Duplex Mode

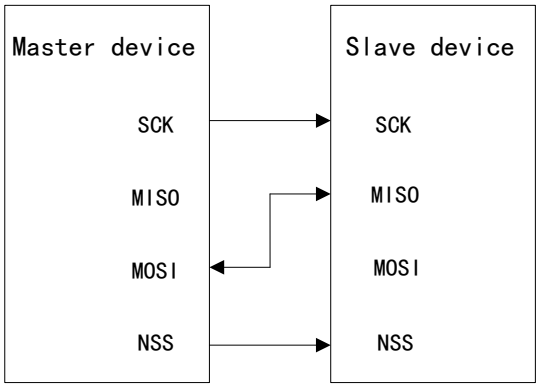


Figure 73 Connection in Simplex Mode (the master is used for receiving data, while the slave is used for transmitting data)

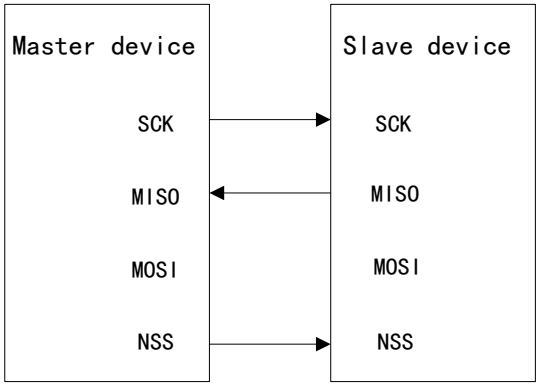
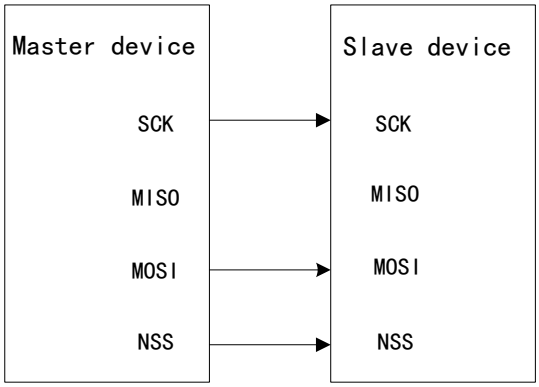


Figure 74 Connection in Simplex Mode (the master only transmits data, while the slave receives data)



19.4.5.1 Transmitting and receiving of data

In order to prevent overrun when the data frame is short and ensure that SPI can work continuously, all SPI data need to pass through the 32-bit embedded FIFO. Each direction will have its own FIFO, TXFIFO and RXFIFO.

Handle FIFO according to SPI simplex and duplex mode, data frame format, access size executed on FIFO data register and whether to use data package to process FIFO when accessing FIFO.

After read access to SPI_DATA register, the earliest values that have not been read yet and are stored in RXFIFO will be returned. After write access to SPI_DATA, the written data will be stored in TXFIFO at the end of the transmit queue. Read access must always be aligned with RXFIFO threshold value configured by FRTCFG bit in SPI_CTRL2 register. The FTLSEL and FRLSEL bits indicate the current occupancy levels of the two FIFO.

The read access to SPI_DATA register must be managed by RXBNEFLG event. When the data are stored in RXFIFO and reach the threshold value (defined by previous bit), this event will be triggered; when RXBNEFLG is cleared, RXFIFO will be regarded to be empty, and in the similar way, the write access to the data frame to be transmitted is managed by TXBEFLG event. When TXFIFO is less than or equal to half of its capacity, RXBNEFLG event will be triggered; otherwise, TXBEFLG will be cleared, meanwhile, it will be regarded that there are data stored in TXFIFO. Therefore, when the data frame format is less than or equal to one byte, RXFIFO can store 4 data frames at most, and TXFIFO can store 3 data frames. When the software attempts to write more data to TXFIFO in 16-bit mode, this difference can prevent the three 8-bit data frames that have been stored in TXFIFO from being damaged. TXBEFLG and RXBNEFLG events can be polled or handled by interrupt.

19.4.5.2 Sequence processing

In transmitting data, multiple data can be formed into a sequence in order.

When the transmission is started, TXFIFO will transmit continuously in order.

In single receive mode, in half-duplex or simplex mode, when SPI is enabled, the master device will immediately receive the sequence until SPI is disabled or the single receive mode is disabled. When the data frame starts transmission, the slave cannot control the data sequence, so the slave must prepare the data before the transmission, to ensure there are data to be transmitted in TXFIFO.

When there are multiple slave devices, each sequence needs to correspond to different slave devices, so NSS pulse should be used to separate the sequence to ensure it is correct.

Note:

(1) Check whether the data transmission is completed according to FTLSEL bit and BSYFLG bit, and the clock output will stop when the transmission is completed.

(2) In packet mode, special attention should be paid to empty bytes when the data being transmitted are odd.

(3) In single receive mode of the master device, it is required to disable SPI or single receive mode to stop clock output.

(4) Master the correct receiving time to ensure the correct data transmission

(5) The action of disabling should be performed between the sampling time of first bit and the first bit of the next byte.

19.4.5.3 Data packing

If the data frame size is less than or equal to one byte, when executing 16-bit read and write access to SPI_DATA register, the data will be packed automatically and double data can be processed in parallel. After conducting write access to SPI_DATA, 2-byte data will be transmitted; if the threshold value of RXFIFO is set to 16 bits, a receive RXBNEFLG event will be generated.

For a single RXBNEFLG event, the data receiver shall perform one read operation to SPI_DATA, and only after that, can it obtain all data.

Note: The threshold value of RXFIFO should be consistent with the bit width of follow-up data access.

19.4.6 NSS pulse mode

NSS pulse mode can be set by configuring NSSPEN bit of SPI_CTRL1 register; this mode takes effect only when SPI is configured as Motorola master mode and captures the first edge. In transmitting of this mode, NSS pulse is generated between two continuous data frames, and NSS will remain high for at least one cycle. NSS pulse mode allows the slave to latch data.

19.4.7 TI mode

Master mode of TI protocol

SPI interface can be made compatible with master mode of TI protocol by configuring FRFCFG bit of SPI_CTRL2 register.

In master mode of TI protocol, it is unaffected by the setting of SPI_CTRL1 register, and the clock polarity, phase and NSS management will meet the requirements of TI protocol. In slave mode, SPI baud rate frequency divider is used to control MISO pin to make MISO pin in high-impedance state, and any baud rate can be used, ensuring the best flexibility.

Generally, the baud rate is set as the baud rate of external master clock, and the delay for MISO signal to become the high-impedance state depends on the baud rate set synchronously and through BRSEL bit of SPI_CTRL1 register internally. The formula is:

$$T_{baud\_rate}/2+4\times t_{pclk}<t_{release}<t_{baud\_rate}+6\times t_{pclk}$$

Note: This function does not apply to Motorola SPI communication mode (FRFCFG bit is set to 0).

19.4.8 CRC Functions

SPI module contains two CRC computing units, which are used for data receiving and data transmission respectively.

CRC computing units are used to define polynomials in SPI_CRCPOLY register (it should be an odd number, and does not support even number).

Enable CRC computing by configuring CRCEN bit in SPI_CTRL1 register; at the same time, reset the CRC register (SPI_RXCRC and SPI_TXCRC).

CRC is managed by CPU during transmission

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 transmits the CRC value after the last data is transmitted, and the CRCNXT bit will be cleared; during CRC data transmission, CRC computing will be frozen.

The received CRC data will be stored in RXFIFO. A CRC transaction usually needs one more data frame to communicate at the end of the data sequence. However, when an 8-bit data frame checked by 16-bit CRC is set, two data frames are needed to transmit the complete CRC. When the last CRC data is received, the received value and the value of SPI_RXCRC register will be compared. By checking CRCEFLG flag bit in SPI_STS register, judge whether the data are destroyed in the process of transmission. CRCEFLG bit can be cleared by writing 0. RXBNEFLG bit can be cleared by reading SPI_DATA register.

Sequence of clearing CRC values

(1) Disable SPI (SPIEN=0)

(2) Clear CRCEN bit

(3) Set CRCEN bit to 1

(4) Enable SPI (SPIEN=1)

Note: When SPI works in slave mode, the software must enable CRC operation when the clock is stable. During the period of the data phase and CRC phase, the NSS signal needs to be pulled down and maintained.

19.4.9 DMA Function

The request/response DMA mechanism in SPI facilitates high-speed data transmission, improves the system efficiency and enable to transfer data to SPI transmit buffer promptly, and the receive buffer can read the data in time to prevent overrun.

When SPI only transmits 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 transmitting: When TXBEFLG flag bit is set to 1, issue the DMA request, DMA controller writes data to SPI_DATA, 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 transmitted in transmitting 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, transmitting and receiving of CRC bytes will be completed automatically. The CRCNXT bit is not controlled by software. The transmitting DMA channel counter of SPI must be set to the number that does not contain CRC data, but the DMA channel counter must contain the length of one more CRC data when receiving.

After reading CRC data in CRC check link, the values of SPI_TXCRC and SPI_RXCRC will be cleared automatically. Then continuous transmission can be realized by DMA circular mode (except in single-receive mode).

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.

19.4.10 Disable SPI

After data transmission is over, end the communication by disabling SPI module.

When data are being transmitted or there are data in TXFIFO, it is not allowed to disable SPI by operating SPIEN bit in SPI_CTRL1 register. If SPIEN=0 is set, the clock signal will be transmitted continuously until the peripheral is enabled again. Certain steps are required to disable SPI to prevent the above situations.

Steps of disabling SPI

(1) Wait for clearing FTLSEL to 0

(2) Wait for clearing BSYFLG flag bit to 0

(3) Wait for clearing FRLSEL to 0

(4) Disable SPI (SPIEN=0)

Steps of disabling SPI in some single-receive mode

(1) Wait for clearing RXOMEN to 0 or setting BMOEN to 1

(2) Wait for clearing BSYFLG flag bit to 0

(3) Wait for clearing FRLSEL to 0

(4) Disable SPI (SPIEN=0)

19.4.11 SPI interrupt

An interrupt can be triggered by the following events during SPI operation:

- TXFIFO ready for loading
- RXFIFO receives data
- Master mode error
- TI frame format error

19.4.11.1 Status flag bit

There are three flag bits for fully monitoring the status of SPI bus:

Transmit buffer idle flag TXBEFLG

TXBEFLG=1 means that TXFIFO has space to store the transmitted data; TXBEFLG flag bit is connected to TXFIFO bit, and in the process of storing data, if the storage content of TXFIFO is less than or equal to FIFO/2, TXBEFLG flag bit remains high. When the storage content of TXFIFO is greater than FIFO/2, TXBEFLG flag bit will be cleared to 0. If TXBEIEN bit in SPI_CTRL2 register is set, an interrupt will be generated.

Receive buffer non-empty flag RXBNEFLG

RXBNEFLG flag bit depends on the value of FRTCFG bit in SPI_CTRL2 register:

- If FRTCFG=1, when the storage content of RXFIFO is greater than or equal to 8 bits, RXBNEFLG=1
- If FRTCFG=1, when the storage content of RXFIFO is greater than or equal to 16 bits, RXBNEFLG=1

RXBNEFLG flag bit will be cleared automatically if not in the above situations.

If RXBNEIEN=1 in SPI_CTRL2 register, an interrupt will be generated.

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. BSYFLG flag can be used to detect whether transmission is over to avoid destroying the last transmitted data.

BSYFLG flag will be cleared in the following situations:

- End the transmission in master mode
- Master mode fault
- In slave mode, there is at least one SPI cycle between two data transmissions
- Disable SPI

During continuous communication:

- 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 data

Note: It is better to use TXBEFLG and RXBNEFLG flags to process the transmitting and receiving of each data item.

19.4.11.2 Error flag bit

Master mode error MEFLG

MEFLG is an error flag bit.

The master mode error occurs when:

- the NSS pin of the master device is pulled down in hardware NSS mode;
- ISSEL bit is cleared in software NSS mode;
- 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, and SPI interface is disabled);
- MSMCFG is cleared and the device is forced to enter the slave mode.

Operation of clearing the MEFLG flag bit:

- When MEFLG bit is set to 1, it is required to read or write SPI_STS register, and then write to SPI_CTRL1 register.

When MEFLG flag bit is 1, it is not allowed to set SPIEN and MSMCFG bits.

Overrun error OVRFLG

An overrun error will be generated when the following events occur:

- When RXBNEFLG flag bit is still 1 after the master device has transmitted data
- When the space in RXFIFO cannot store the data to be received when receiving data
- When the software or DMA cannot read the data in RXFIFO in time
- When CRC is only enabled in receive mode, RXFIFO is not available and the receive buffer is limited to the single data frame buffer

When an overrun error occurs: OVRFLG bit is set to 1; if ERRIEN bit is also set, an interrupt will be generated.

After an overrun error occurs, the data in the receiving buffer are not the data transmitted by the master device, and by reading SPI_DATA value, the data are the data not read before, and the subsequent data will be discarded.

OVRFLG flag can be cleared by reading SPI_DATA register and SPI_STS register according to the sequence.

CRC error flag bit CRCEFLG

Enable CRC operation by setting CRCEN bit of SPI_CTRL1 register, and CRC error flag can be used to check whether the received data are valid.

When the value transmitted 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.

The SPI error interrupt does not respond to the CRC error flag bit. In other words, setting the CRCEFLG (CRC Error Flag) will not generate an interrupt.

TI mode frame format error (FREFLG)

Under the slave device and in accordance with TI mode protocol, when a pulse appears in NSS during data communication, a TI mode frame format error will be caused. When TI mode frame format error occurs, FREFLG flag bit of SPI_STS register will be set to 1, SPI will not be disabled, NSS pulse will be

ignored, and SPI will wait for the next NSS pulse before retransmission. As the error detection may cause the loss of two data bytes, the data may have been destroyed.

FREFLG flag can be cleared by reading SPI_STS register. If ERRIEN bit is set, an interrupt will be generated when NSS error occurs. At this time, SPI is disabled, which is because the consistency of data cannot be guaranteed. When SPI is enabled again, the master server needs to be reinitialized.

Table 63 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	ERRIEN	Read/Write SPI_STS register, and then write SPI_CTRL1 register
OVRFLG	Overrun error		Read SPI_DATA register, and then read SPI_STS register
FREFLG	TI mode frame format error		Read SPI_STS register

19.5 Register Address Mapping

Table 64 SPI Register Address Mapping

Register name	Description	Offset address
SPI_CTRL1	SPI control register 1	0x00
SPI_CTRL2	SPI control register 2	0x04
SPI_STS	SPI status 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

19.6 Register Functional Description

These peripheral registers can be operated by half word (16 bits) or word (32 bits).

19.6.1 SPI control register 1 (SPI_CTRL1)

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 the first clock 1: On the edge of the second clock Note: This bit cannot be modified during communication. Except that CRC function is used in TI mode, this bit is not used in SPI TI mode.
1	CPOL	R/W	Clock Polarity Configure The state maintained by SCK when SPI is in idle state. 0: SCK low 1: SCK high Note: This bit cannot be modified during communication. Except that CRC function is used in TI mode, this bit is not used in SPI TI mode.
2	MSMCFG	R/W	Master/Slave 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=Fmaster/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, operate according to the process of disabling SPI.
7	LSBSEL	R/W	LSB First Transfer Select 0: First transmit the most significant bit (MSB) 1: First transmit the least significant bit (LSB)
8	ISSEL	R/W	Internal Slave Device Select Determine the level on NSS pin This bit can be set effectively only when CTRL1_SSEN=1.
9	SSEN	R/W	Software Slave Device Enable 0: Disable 1: Enable When SSEN is set, the level of NSS pin is determined by SSEN.

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 collision, it is necessary to set RXOMEN bit to 1 on the slave devices that are not accessed.
11	CRCLSEL	R/W	CRC Length Select 0: Use 8-bit CRC 1: Use 16-bit CRC Note: This bit can be written only when SPIEN=0; otherwise, an error will occur.
12	CRCNXT	R/W	Enable Next Transmitted Value CRC 0: Next value to be transmitted is from transmitter buffer 1: Next value to be transmitted is from transmitter 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 (receive-only ode) 1: Enable (transmit-only mode) When BMEN=1, namely in single-line bidirectional mode, this bit decides the transmission direction of transmission line.
15	BMEN	R/W	Bidirectional Mode Enable 0 : Double-line bidirectional mode 1: Single-line bidirectional mode Single-line bidirectional transmission means: transmission between MOSI pin of data master and MISO pin of slave.

19.6.2 SPI control register 2 (SPI_CTRL2)

Offset address: 0x04

Reset value: 0x0700

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	Enable SS Output SS output in master mode 0: Disable SS output, and it can work in multi-master mode. 1: Enable SS output, and it cannot work in multi-master mode. Note: Not available in TI mode of SPI.
3	NSSPEN	R/W	NSS Pulse Management Enable 0: Disable 1: Enable Note: During continuous transmission, it is allowed to generate NSS pulse between transmission of two data. During single data transmission, NSS pin will be forced to be pulled up at the end of transmission. This bit is invalid when CPHA=1 or FRFCFG=1. This bit can be written only when SPIEN=0. Not available in TI mode of SPI.
4	FRFCFG	R/W	Configure Frame Format 0 : SPI Motorola mode 1: SPI TI mode Note: This bit can be written only when SPIEN=0.
5	ERRIEN	R/W	Error interrupt Enable 0: Disable 1: Enable Enabling the ERRIEN error interrupt will trigger the CRCEFLG error flag, but no interrupt will be generated.
6	RXBNEIEN	R/W	Receive Buffer Not Empty Interrupt Enable 0: Disable 1: Allowed 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

Field	Name	R/W	Description
11:8	DSCFG	R/W	Data Size Configure Configure the bit width of SPI transmission date: 0000: Reserved 0001: Reserved 0010: Reserved 0011: 4 bits 0100: 5 bits 0101: 6 bits 0110: 7 bits 0111: 8 bits 1000: 9 bits 1001: 10 bits 1010: 11 bits 1011: 12 bits 1100: 13 bits 1101: 14 bits 1110: 15 bits 1111: 16 bits Note: When reserved bit is written by software, the value will be forced to be 0111 (8 bits)
12	FRTCFG	R/W	FIFO Reception Threshold Configure Configure FIFO threshold, and when the value exceeds this threshold, RXBNEFLG will occur 0: 16 bits 1: 8 bits
13	LDRX	R/W	Last DMA Receive These bits are used in data packing mode to define the total number received by DMA to be odd or even. 0: Even 1: Odd Note: These bits are meaningful only when RXDEN bit of SPI_CTRL2 register is set and the packing mode is enabled. This bit can be written only when SPIEN=0. The SPI is disabled according to the procedure described in the section of “Disabling SPI”.
14	LDTX	R/W	Last DMA Transmit These bits are used in data packing mode to define the total number transmitted by DMA to be odd or even. 0: Even 1: Odd Note: These bits are meaningful only when RXDEN bit of SPI_CTRL2 register is set and the packing mode is enabled. This bit can be written only when SPIEN=0. The SPI is disabled according to the procedure described in the section of “Disabling SPI”.

Field	Name	R/W	Description
15	Reserved		

19.6.3 SPI status register (SPI_STS)

Offset address: 0x08

Reset value: 0x0002

Field	Name	R/W	Description
0	RXBNEFLG	R	Receive Buffer Not Empty Flag This bit indicates that the receive buffer is empty or not 0: Empty 1: Not empty
1	TXBEFLG	R	Transmit Buffer Empty Flag This bit indicates that the transmit buffer is empty or not 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.
3	UDRFLG	R	Underrun Occur Flag This bit indicates whether the underrun occurs or not 0: Not occurred 1: Occurred This flag bit is set by hardware and reset 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 and reset by software.
5	MEFLG	R	Mode Error Occur Flag This bit indicates whether mode error occurs or not 0: Not occurred 1: Occurred This bit can be set by hardware and reset by software.
6	OVRFLG	R	Overrun Occur Flag This bit indicates whether overrun occurs or not 0: Not occurred 1: Occurred This bit can be set by hardware and reset by software.

Field	Name	R/W	Description
7	BSYFLG	R	Busy Flag This bit indicates the work state of SPI 0: SPI is idle 1: SPI is communicating This bit can be set or reset by hardware
8	FREFLG	R	Frame Format Error Flag 0: Not occurred 1: Occurred Note: This bit is set to 1 by hardware and cleared by reading SPI_STS register.
10:9	FRLSEL	R	FIFO Receive Level Select p00: FIFO is empty 01: FIFO/4 10: FIFO/2 11: FIFO is full Note: This bit is set to 1 or cleared by hardware. It is not used in SPI single receive mode with CRC check.
12:11	FTLSEL	R	FIFO Transmit Level Select p00: FIFO is empty 01: FIFO/4 10: FIFO/2 11: FIFO is full (it can be considered as full when the threshold value of FIFO is greater than 1/2) Note: This bit is set to 1 or cleared by hardware.
15:13	Reserved		

19.6.4 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 Store the data to be transmitted or received. 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, DATA[7:0] will be used when transmitting and receiving data, and DATA[15:8] is invalid; for 16-bit data, DATA[15:0] will be used when transmitting and receiving data.

19.6.5 SPI CRC polynomial register (SPI_CRCPOLY)

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, which can be modified and the reset value is 0x0007.

19.6.6 SPI Receiving CRC Register (SPI_RXCRC)

Offset address: 0x14

Reset value: 0x0000

Field	Name	R/W	Description
15:0	RXCRC	R	Receive Data CRC Value The CRC data of receive bytes 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.

19.6.7 SPI Transmitting 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 bytes calculated by hardware are stored in TXCRC; the bits and the length of data frames are consistent, that is, if the transmitted data are 8 bits, the CRC computing is made based on CRC8; if the transmitted data is are 16 bits, the CRC computing is made based on CRC16. Note: When BSYFLG bit is set to 1, the value of reading TXCRC register may be wrong.

20 Controller Area Network (CAN)

20.1 Full Name and Abbreviation Description of Terms

Table 65 Full Name and Abbreviation Description of Terms

Full Name	Abbreviation
Bit Rate Prescaler	BRP
Bit Stream Processor	BSP
Bit Timing Logic	BTL
Controller Area Network	CAN
Controller Area Network with Flexible Data-rate	CAN FD
Cyclic Redundancy Check	CRC
Data Length Code	DLC
Error Correction Code	ECC
Electronic Control Unit	ECU
Error Management Logic	EML
End of Frame	EOF
Finite State Machine	FSM
minimum time quantum = CAN clock period (can_cclk)	mtq
Start of Frame	SOF
Secondary Sample Point	SSP
Transmitter Delay Compensation	TDC
Time quantum	Tq
Time Segment before Sample Point	TSEG1
Time Segment after Sample Point	TSEG2

20.2 Introduction

The CAN module supports the standard of ISO 11898-1:2015 and the protocol of BOSCH CAN FD (Version: V1.0). To be connected to the physical layer, an additional transceiver hardware is required.

The Message RAM is implemented as a single-port SRAM and can be accessed by either the CPU or the CAN module.

All message handling-related functions are implemented by the Receive Handler (Rx Handler) and the Transmit Handler (Tx Handler). The Receive Handler manages message reception filtering, transfers received messages

from the CAN core to the Message RAM, and provides the status information on received messages. The Transmit Handler is responsible for transferring messages to be sent from the Message RAM to the CAN core and provides transmission status information.

The reception filtering function is achieved through up to 128 filter combinations. Each filter can be configured as a range filter, a bit mask filter, or a specific ID filter.

The CAN module can be connected to the CPU via a 32-bit generic slave interface. The clock domain of the CAN module allows for the high-precision CAN clock to be separated from the host clock which can be generated by a PLL.

20.3 Main Characteristics

(1) Conforms to ISO 11898-1:2015

(2) Supports the CAN FD protocol with data payloads of up to 64 bytes

(3) Features CAN error logging functionality

(4) Supports AUTOSAR (Automotive Open System Architecture)

(5) Supports the SAE J1939 protocol (a communication protocol for commercial vehicles)

(6) Improved acceptance filtering capability

(7) Two configurable receive First-In-First-Out (FIFO) buffers

(8) Provides a dedicated signal upon reception of a high-priority message

(9) Up to 64 dedicated receive buffers

(10) Up to 32 dedicated transmit buffers

(11) Configurable transmit FIFO buffers

(12) Configurable transmit queue

(13) Configurable transmit event FIFO buffer

(14) CPU has direct access to the Message RAM

(15) Programmable loopback test mode

(16) Maskable module interrupts

(17) Dual clock domains (CAN protocol clock and CAN peripheral clock)

(18) Supports power-down mode and wake-up functionality

(19) Supports SRAM ECC check; ECC errors can trigger an interrupt and the error address can be read

20.4 Structure Block Diagram

Figure 75 CAN Controller Structure Block Diagram

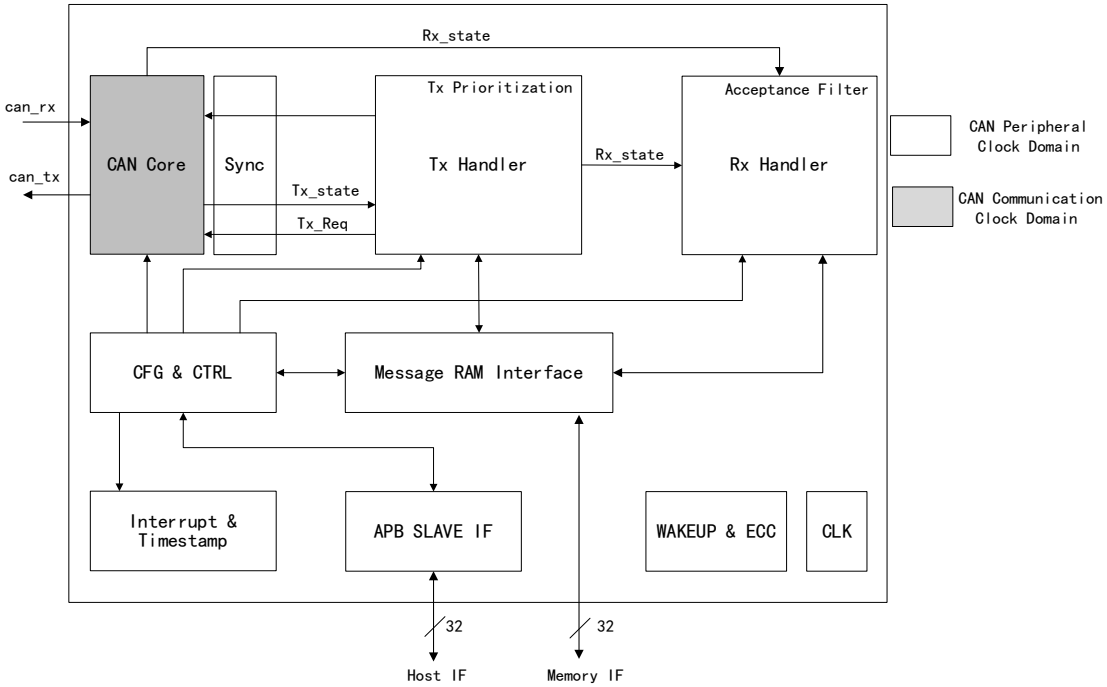


Table 66 Description of CAN Controller Structure Block Diagram

Name	Meaning	Description
CAN Core	CAN core	The CAN protocol controller and the Receive/Transmit Shift Registers are responsible for handling all protocols compliant with the ISO 11898-1:2015, supporting both 11-bit and 29-bit identifiers.
Sync	Synchronization	It synchronizes signals from the CAN peripheral clock domain to the CAN protocol clock domain, and vice versa.
CLK	Clock	It synchronizes the CAN reset signal to both the CAN peripheral clock domain and the CAN protocol clock domain.
CFG & CTRL	Configuration and control	This includes configuration and control bits related to the CAN core.
Interrupt & Timestamp	Interrupt and timestamp	It features interrupt control and a 16-bit CAN bit time counter for generating receive and transmit timestamps. Alternatively, an externally generated 16-bit vector can be used instead of the internal 16-bit CAN bit time counter to generate these timestamps.

Name	Meaning	Description
Tx Handler	Transmit Handler	It controls the transfer of messages from the external message RAM to the CAN core. Up to 32 transmit buffers can be configured for message transmission. These buffers can be used as dedicated transmit buffers, as part of a transmit FIFO buffer, as part of a transmit queue, or in a combination of these modes. The Transmit Event FIFO stores the transmit timestamp along with the corresponding message ID. Transmit abort functionality is also supported.
Rx Handler	Receive Handler	It controls the transfer of received messages from the CAN core to the external message RAM. The Receive Handler supports two configurable Receive FIFO buffers, as well as up to 64 dedicated receive buffers for storing all messages that pass the acceptance filtering. Unlike the Receive FIFO buffers, dedicated receive buffers are used to store only messages with specific identifiers. Each message is stored along with a receive timestamp. A maximum of 128 filters can be defined for 11-bit ID, and up to 64 filters for 29-bit ID.
APB Slave Interface	APB Slave Interface	The CAN module is connected to the APB bus. Its APB Slave Interface supports connection only to a 32-bit bus.
Message RAM Interface	Message storage interface	The external 32-bit message RAM is accessible by both the CPU and the CAN controller.
Wakeup & ECC	Wake-up and ECC check	The CAN module supports waking the system from a low-power mode (STOP mode) and also features ECC check.
Can_pclk & can_cclk	Double clock source	The CAN module operates with two clock domains: the CAN peripheral clock (can_pclk) and the CAN protocol clock (can_cclk), which correspond to the APB CLK and CAN clock described in the section of “Clock Tree”, respectively.
Interrupt	Interrupt	The module provides three interrupt lines. Interrupt signals can be routed to CAN_INT0, CAN_INT1, and CAN_SMS_IT. For the CAN controller, all interrupts can be routed to the CAN_INT0 and CAN_INT1 lines. These two interrupt lines can be individually enabled or disabled by programming the ILE[EINT0] and ILE[EINT1] bits, respectively. For wake-up and ECC check events, all interrupts can be routed to the CAN_SMS_IT interrupt line.

20.5 Functional Description

20.5.1 Operation mode

20.5.1.1 Software initialization

Software initialization is triggered by setting the CCCR [INIT] bit. It can also be initiated by a software setting, a hardware reset, or when the CAN controller enters the Bus-Off state. When CCCR [INIT] bit is set, message transmission on the CAN bus is halted, and the CAN bus output can_tx becomes recessive (high level). The counter of the Error Management Logic (EML) remains unchanged. Setting CCCR [INIT] does not alter any configuration registers. Resetting CCCR [INIT] completes the software initialization.

Following this, the Bit Stream Processor (BSP) must synchronize with the data transmission on the CAN bus by waiting for the occurrence of 11 consecutive recessive bits (indicating bus idle, Bus_Idle). Only after this synchronization can it participate in bus activities and begin message transmission.

Access to the CAN configuration registers is permitted only when both the CCCR [INIT] and CCCR [CCE] bits are set (via a protected write).

The CCCR [CCE] bit can only be set or reset when CCCR [INIT] = 1. When CCCR [INIT] is reset, CCCR [CCE] is automatically reset.

When the CCCR [CCE] bit is set, the following registers are reset:

- HPMS (High Priority Message Status)
- RXF0S (Rx FIFO 0 Status)
- RXF1S (Rx FIFO 1 Status)
- TXFQS (Tx FIFO/Queue Status)
- TXBRP (Tx Buffer Request Pending)
- TXBTO (Tx Buffer Transmission Occurred)
- TXBCF (Tx Buffer Cancellation Finished)
- TXEFS (Tx Event FIFO Status)

When CCCR [CCE] is set, the timeout counter value TOCV [TOC] is preloaded with the value configured in TOCC [TOP].

Furthermore, when CCCR [CCE] = 1, the state machines of the Tx Handler and the Rx Handler remain in an idle state.

The following registers are writable only when CCCR [CCE] = 0:

- TXBAR (Tx Buffer Add Request)
- TXBCR (Tx Buffer Cancellation Request)

The CCCR [TEST] and CCCR [MON] bits can only be set by the host when CCCR [INIT] = 1 and CCCR [CCE] = 1. Both bits can be reset at any time. The CCCR [DAR] bit can only be set/reset when CCCR [INIT] = 1 and CCCR [CCE] = 1.

Note: The message RAM is equipped with ECC functionality. It is recommended to initialize the message RAM after a hardware reset by writing, for example, 0x0000 0000 to each message RAM word to generate a valid ECC checksum. This prevents the activation of ECC interrupts when reading from uninitialized sections of the message RAM.

20.5.1.2 Normal operation

Once the CAN controller completes initialization and the CCCR [INIT] bit is reset to 0, it synchronizes with the CAN bus and is ready for communication.

- The received messages (including the DLC and message ID) are stored into either a dedicated Rx Buffer, Rx FIFO 0, or Rx FIFO 1 after passing the acceptance filtering.
- The messages to be transmitted can be used for the initialization or update of the dedicated Tx Buffers, Tx FIFO, or Tx Queue. Automatic transmission of remote frames is not supported.

20.5.1.3 CAN FD operation

There are two variants of CAN FD frame transmission. The first is a CAN FD frame with bit rate switching disabled. The second is a CAN FD frame where the Control Field, Data Field, and CRC Field are transmitted at a faster bit rate compared to the beginning and end parts of the frame.

In a CAN frame with an 11-bit identifier, the previously reserved bits, and the first reserved bit in a CAN frame with a 29-bit identifier, will now be decoded as the FDF (FD Frame) bits. FDF recessive indicates a CAN FD frame, while FDF dominant indicates a classic CAN frame.

In a CAN FD frame, the two bits following FDF, res (reserved) and BRS (Bit Rate Switch), determine whether the bit rate is switched within this CAN FD frame. CAN FD bit rate switching is indicated by res = dominant and BRS = recessive. The recessive (res) code is reserved for future protocol extensions. If the CAN controller receives a frame with FDF = recessive and res = recessive, it signals a Protocol Exception Event by setting the PSR[PXE] (Protocol Exception Event) bit. When protocol exception handling is enabled (CCCR [PXHD] = 0), this causes the module's active state to change from Receiver (PSR[ACT] = 10) to Synchronizing (PSR[ACT] = 00) at the next sampling point. When protocol exception handling is disabled (CCCR [PXHD] = 1), the CAN controller treats the recessive res bit as a format error and responds with an error frame.

CAN FD operation is enabled by programming the CCCR [FDOE] bit.

If CCCR [FDOE] = 1, the transmission and reception of CAN FD frames are enabled. The controller can always transmit and receive classic CAN frames. It determines whether a CAN FD frame or a classic CAN frame is transmitted according to the configuration of the FDF bit in the corresponding Tx Buffer Element.

If CCCR [FDOE] = 0, received frames are interpreted as classic CAN frames. This will cause an error frame to be transmitted if a CAN FD frame is received. When CAN FD operation is disabled, CAN FD frames are not transmitted, even if the FDF bit in the Tx Buffer Element is set. The CCCR [FDOE] and CCCR [BRSE] bits can only be changed when both CCCR [INIT] and CCCR [CCE] are set.

If CCCR [FDOE] = 0, the settings of the FDF and BRS bits are ignored, and the frames are transmitted in the format of classic CAN. If CCCR [FDOE] = 1 and CCCR [BRSE] = 0, only the FDF bit in the Tx Buffer Element is evaluated. If CCCR [FDOE] = 1 and CCCR [BRSE] = 1, the transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer Elements that have both the FDF and BRS bits set are transmitted in the CAN FD format with bit rate switching.

During CAN operation, a mode change is recommended only under the following circumstances:

(1) During system startup, all nodes transmit classic CAN frames until it is verified that they are capable of communicating in the CAN FD format. If so, all nodes switch to CAN FD operation.

(2) Wake-up messages in a partial CAN network must be transmitted in the classic CAN format.

(3) The fault rate during the CAN FD data phase is significantly higher than that during the CAN FD arbitration phase. In this case, the CAN FD bit rate switching option for transmission is disabled.

(4) End-of-line programming is implemented if not all nodes support CAN FD operation. Non-CAN FD nodes remain in mute mode until programming is completed. Afterwards, all nodes switch back to classic CAN communication mode.

In the CAN FD format, the DLC code differs from the standard CAN format. DLC codes 0 to 8 are encoded the same as in standard CAN, while codes 9 to 15, which are all encoded as an 8-byte data field in standard CAN, are encoded according to the following table in CAN FD.

Table 67 DLC Codes

DLC	9	10	11	12	13	14	15
Number of data bytes	12	16	20	24	32	48	64

In a CAN FD frame, if the BRS (Bit Rate Switch) bit is recessive, the bit timing switches within the frame after this bit. Before the BRS bit, the nominal CAN bit timing defined by the nominal bit timing and Prescaler Register (NBTP) is used during the CAN FD arbitration phase. In the subsequent CAN FD data phase, the data phase bit timing defined by the Data Bit Timing and Prescaler Register

(DBTP) is used. The bit timing switches back from the data phase timing at the CRC delimiter or upon error detection, whichever occurs first.

The maximum configurable bit rate for the CAN FD data phase depends on the CAN protocol clock frequency (can_cclk). For example: If the CAN protocol clock frequency is 20 MHz and the minimum configurable bit time is 4 tq, the resulting bit rate for the data phase is 5 Mbit/s.

In both data frame formats (CAN FD and CAN FD with bit rate switching), the value of the ESI (Error State Indicator) bit is determined by the error state of the transmitter at the start of the transmission. If the transmitter is in the error-passive state, the ESI bit is transmitted in recessive format; otherwise, it is transmitted in dominant format.

20.5.1.4 Transmitter delay compensation

During the data phase of a CAN FD transmission, only one node acts as the transmitter, while all other nodes are receivers. The bus length has no impact in this phase. When transmitting via the can_tx pin, the CAN controller simultaneously receives the transmitted data from the local CAN transceiver via the can_rx pin. The received data is delayed due to transmitter loop delay. If this delay exceeds TSEG1 (the time segment before the sampling point), a bit error will be detected. To achieve data phase timings shorter than this transmitter loop delay, a transmitter delay compensation (TDC) mechanism is introduced. Without transmitter delay compensation, the achievable bit rate in the data phase of a CAN FD frame would be limited by the transmitter loop delay.

Description

The protocol unit of the CAN controller implements a Transmitter Delay Compensation (TDC) mechanism to compensate for the transmitter loop delay, thereby enabling higher bit rates in the CAN FD data phase without being limited by the specific delay of a CAN transceiver.
To check for bit errors in the data phase at the transmitting node, the delayed transmitted data is compared with the received data at a Secondary Sampling Point (SSP). If a bit error is detected, the transmitter will react to this bit error at the next regular sampling point. The delay compensation is always disabled during the arbitration phase.

The transmitter delay compensation allows for the configuration of data bit times that are shorter than the transmitter loop delay, as detailed in ISO 11898-1:2015. This feature is enabled by setting the DBTP [TDC] bit.

At the SSP, the receiving bit is compared with the transmitting bit. The SSP position is defined as the sum of the measured loop delay from the controller's transmit output can_tx through the transceiver to its receive input can_rx, and the configured Transmitter Delay Compensation Offset TDCR [TDCO]. This offset is used to adjust the position of the SSP within the receiving bit (for

example, to half of the data phase bit time). The calculated position of the secondary sampling point is rounded down to the next integer multiple of the minimum time quantum (mtq).

The PSR [TDCV] register displays the actual measured transmitter delay compensation value. PSR [TDCV] is cleared when CCCR [INIT] is set and is updated each time an FD frame is transmitted while DBTP [TDC] is set.

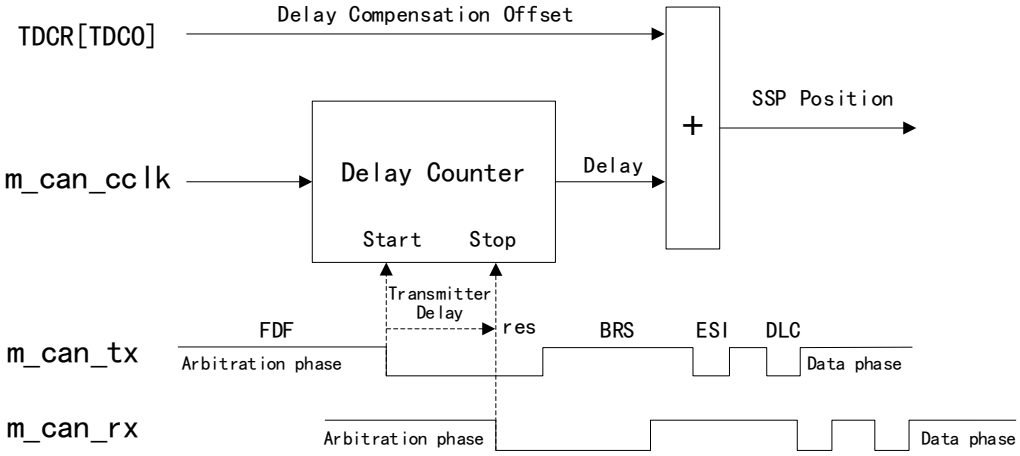
The transmitter delay compensation mechanism in the CAN controller must adhere to the following boundary conditions:

- The sum of the measured loop delay from can_tx to can_rx and the configured transmitter delay compensation offset TDCR[TDCO] must be less than or equal to 127 mtq. If this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- During the data phase, the sum of the measured loop delay from can_tx to can_rx and the configured transmitter delay compensation offset TDCR[TDCO] must be less than 6-bit time.
- The data phase ends at the sampling point of the CRC delimiter, at which point the checking of receiving bits at the SSP ceases.

Transmitter delay compensation measurement

If transmitter delay compensation is enabled by programming the DBTP[TDC] bit to 1, the measurement starts at the falling edge from the FDF bit to the res bit in each transmitted CAN FD frame. The measurement stops when this edge is detected at the transmitter's receive input, can_rx. The resolution of this measurement is one minimum time quantum (mtq).

Figure 76 Transmitter Delay Measurement



To prevent a dominant glitch within the received FDF bit from prematurely ending the delay compensation measurement before the falling edge of the received res bit – which would cause the SSP to be positioned too early – a transmitter delay compensation filter window can be enabled by programming

TDCR [TDCF]. This defines a minimum value for the SSP position. For the transmitter delay measurement, dominant edges on can_rx that would result in an earlier SSP position are ignored. The measurement stops when the calculated SSP position is at least TDCR [TDCF] and can_rx is at a low level.

Note: The CAN controller always performs a dynamic transmitter delay measurement; the configuration of static (fixed) transmitter delay is not supported.

20.5.1.5 Restricted operation mode

In Restricted Operation Mode, a node is capable of receiving data and remote frames and acknowledging valid frames, but it will not transmit data frames, remote frames, active error frames, or overload frames. In the event of an error condition or overload condition, instead of transmitting dominant bits, it waits for the occurrence of a bus idle state to resynchronize with the CAN communication. While error logging (ECR[CEL]) is active, the error counters (ECR[REC], ECR[TEC]) are frozen. The host can set the CAN controller into Restricted Operation Mode by setting the CCCR[ASM] bit. This bit can only be set by the host when both CCCR [CCE] and CCCR [INIT] bits are set to 1. The host can reset this bit at any time.

The controller automatically enters Restricted Operation Mode when the Tx Handler fails to read data from the message RAM in time. To exit Restricted Operation Mode, the host CPU must reset CCCR [ASM].

Restricted Operation Mode can be used in applications that need to adapt to different CAN bit rates. In such cases, the APP tests different bit rates and exits Restricted Operation Mode upon receiving a valid frame.

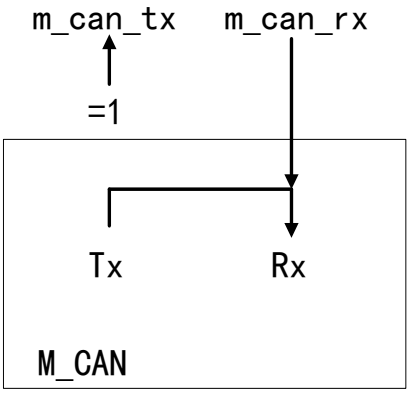
Note: Restricted Operation Mode must not be used in combination with Loopback Mode (internal or external).

20.5.1.6 Bus monitoring mode

The CAN controller can be set to Bus Monitoring Mode by programming the CCCR [MON] bit to 1. In Bus Monitoring Mode (see “10.14 Bus Monitoring” in the ISO 11898-1:2015), the CAN controller is able to receive valid data frames and valid remote frames but cannot initiate a transmission. In this mode, it transmits only recessive bits on the CAN bus. If the controller needs to send a dominant bit (such as an ACK bit, overload flag, or active error flag), this bit is internally rerouted so that the controller can monitor it, even though the actual CAN bus may remain in the recessive state. In Bus Monitoring Mode, the TXBRP register remains in its reset state.

Bus Monitoring Mode can be used to analyze the traffic on the CAN bus without affecting it by transmitting dominant bits. The figure below shows the connection of the can_tx and can_rx signals when the CAN controller is in Bus Monitoring Mode.

Figure 77 Bus Monitoring Mode



20.5.1.7 Disable automatic retransmission

According to the CAN specification (see “8.3.4 Recovery Management” in the ISO 11898-1:2015), the CAN controller provides a method for the automatic retransmission of frames that have lost arbitration or have been disturbed by errors during their transmission. By default, automatic retransmission is enabled. To support the time-triggered communication as described in Section 9.2 of ISO 11898-1:2015, automatic retransmission can be disabled via the CCCR [DAR] bit.

Frame transmission in DAR mode

In DAR (Disable Automatic Retransmission) mode, all transmissions initiated on the CAN bus are automatically cancelled. Upon a successful transmission, the Tx request pending bit TXBRP [TRPx] in the Tx Buffer will be cleared; this bit is also cleared if the transmission has not started by the cancellation point, is aborted due to arbitration loss, or if an error occurs during frame transmission.

- When arbitration is lost or the frame transmission is disturbed, the corresponding Tx Buffer Transmission Occurred bit TXBTO [TOx] is not set, and the corresponding Tx Buffer Cancellation Finished bit TXBCF[CFx] is set.
- When transmission is successful, the corresponding Tx Buffer Transmission Occurred bit TXBTO[TOx] is set, and the corresponding Tx Buffer Cancellation Finished bit TXBCF[CFx] is not set.
- When a frame is successfully transmitted despite cancellation, the corresponding Tx Buffer Transmission Occurred bit TXBTO [TOx] is set, and the corresponding Tx Buffer Cancellation Finished bit TXBCF[CFx] is set.

If a frame is successfully transmitted and Tx Event storage is enabled, an event with Event Type ET = 10 (transmission occurred despite cancellation) is written into the Tx Event FIFO element.

20.5.1.8 Power-down (sleep mode)

The controller enters power-down mode via an externally input clock stop request signal or by setting the CCCR [CSR] bit. The CCCR [CSR] bit reads as 1 as long as the external clock stop request signal is active.

After all pending transmission requests are completed, the CAN controller waits until a bus idle state is detected. It then sets CCCR [INIT] bit to 1 to prevent any further CAN transmissions. Next, the CAN controller acknowledges it is ready for power-down by setting the output clock stop acknowledge signal to 1 and setting the CCCR[CSA] bit to 1. Further register accesses are possible before deactivating the module clocks can_hclk and can_cclk, with the exception of the CCCR[INIT] bit, which remains set to 1.

Note: In cases of severe disturbance on the CAN bus, a bus idle state might not be reached, and consequently, the CAN controller will not set CCCR [INIT] bit. This condition can be detected by polling the PSR[ACT] register. If the CAN controller does not enter the idle state, software can write CCCR [INIT] = 1, which will immediately halt the controller's CAN communication, regardless of any ongoing transmission or reception.

To exit power-down mode, the APP must open the module clock before resetting the external clock stop request signal or the CCCR [CSR] bit. The CAN controller will acknowledge this action by resetting the output clock stop acknowledge signal and clearing the CCCR[CSA] bit. Then the APP restarts CAN communication by resetting CCCR [INIT] bit.

20.5.1.9 Wake-up request

When the system is in STOP mode, the CAN module generates a wake-up request upon the occurrence of the following wake-up event:

- A dominant bit (logic 0) on the can_rx pin

The user can wake the system from STOP mode either via an interrupt (by setting ERM_CTRL[WAKE_INT_EN]) or via an event (by setting ERM_CTRL[WAKE_EVENT_EN]).

Note: After the CAN module wakes the system from STOP mode, to restart CAN communication the user must enable the HSE clock and then reset CCCR[INIT].

20.5.1.10 Test Mode

Write access to the TEST register is only permitted when the CCCR[TEST] bit is set to 1. This enables the configuration of test mode and test functions.

By programming the TEST[TX] bits, the CAN transmit pin can_tx can be configured for one of four output functions. In addition to its default function as a serial data output, it can be driven to output the CAN sampling point signal for monitoring the CAN bit timing, or to output a constant dominant or recessive value. The current logic level of the receive pin can_rx can be read from the TEST[RX] bit. These two functions can be used to check the physical layer of

the CAN bus.

Due to the synchronization mechanism between the CAN communication clock and the CAN peripheral clock domains, a latency of a few host clock cycles may occur from writing to TEST[TX] bit until the newly configured value appears on the output pin can_tx. The same applies when reading the input pin can_rx via the TEST[RX] bit.

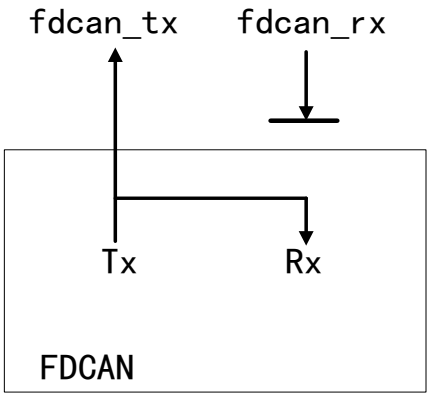
Note: The test mode should only be applicable for production testing or self-test. Software control of the can_tx pin interferes with all CAN protocol functions. The test mode is not recommended to be used in applications.

External Loopback Mode

The CAN controller can be set to External Loopback Mode by programming the TEST[LBCK] bit to 1. In Loopback Mode, the CAN controller treats the messages it transmits as received messages and stores them (if they pass acceptance filtering) into the Rx Buffer or Rx FIFO. The diagram below illustrates the connection of the signal input pins can_tx and can_rx to the CAN controller in External Loopback Mode.

This mode is used for hardware self-test. To remain independent of external influences, the CAN controller ignores acknowledge errors (sampling a recessive bit in the acknowledge slot of a data/remote frame) in Loopback Mode. In this mode, the CAN controller internally feeds back its Tx output to its Rx input. The actual value of the can_rx input pin is ignored by the CAN controller. The transmitted messages can be monitored on the can_tx pin.

Figure 78 External Loopback Mode

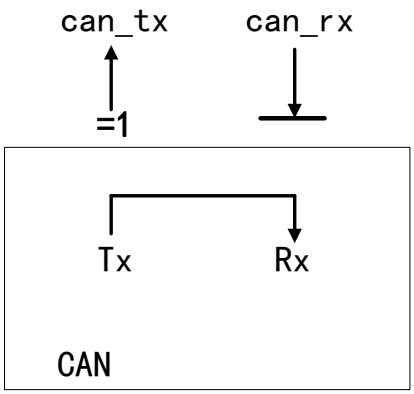


Internal Loopback Mode

The CAN controller can enter Internal Loopback Mode by setting the TEST[LBCK] and CCCR[MON] bits to 1. This mode can be used for a “hot self-test,” allowing the CAN controller to be tested without affecting the operational CAN system connected to the can_tx and can_rx pins. In this mode, the can_rx pin is disconnected from the CAN controller, while the can_tx pin is held at the

recessive state. The diagram below illustrates the connection of the can_tx and can_rx pins to the CAN controller in Internal Loopback Mode.

Figure 79 Internal Loopback Mode



20.5.2 Timestamp generation

20.5.2.1 Internal timestamp generation

The CAN controller implements an internal timestamp generation mechanism using a 16-bit wrapping counter. This counter can be configured via the prescaler TSCC[TCP] to increment at multiples (1…16) of the CAN bit time. The current value of the counter can be read from the TSCV[TSC] register. Writing to the TSCV register resets the counter to zero. The interrupt flag IR[TSW] is set whenever the timestamp counter wraps around.

At the start of a frame's reception or transmission (SOF), the value of the counter is captured. This captured value is stored in the timestamp section (for example, RXTS[15:0]) of the respective Rx buffer, Rx FIFO element, or Tx Event FIFO element (for example, TXTS[15:0]).

20.5.2.2 External timestamp generation

As an alternative to using the internal 16-bit counter, an external 16-bit time base vector input of CAN can be captured as the timestamp by programming the TSCC[TSS] bits.

The external timestamp clock source is provided by TMR8. For detailed usage of TMR8, refer to the specific descriptions in the TMR8 Manual.

20.5.3 Timeout counter

The CAN controller is provided with a 16-bit timeout counter to signal timeout for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO. The timeout counter is configured via the TOCC register, and its current value can be read from TOCV[TOC].

This timeout counter operates as a decrementing counter and uses the same

prescaler controlled by TSCC[TCP] as the timestamp counter. The timeout counter can only start when CCCR[INIT] = 0. It stops when CCCR[INIT] = 1, for example, when the CAN controller enters the Bus-Off state.

The interrupt flag IR[TOO] is set when the value of the counter reaches zero. In continuous mode, the counter immediately restarts from the value configured in TOCC[TOP].

The operating mode is selected via TOCC[TOS]. When operating in continuous mode, the counter begins running when CCCR[INIT] is reset. Writing to the TOCV register presets the counter to the value configured in TOCC[TOP], and it continues decrementing.

When the timeout counter is controlled by a specific FIFO, an empty FIFO causes the counter to be preset to the value configured in TOCC[TOP]. Decrementing begins when the first FIFO element is stored. In this mode, writing to the TOCV register has no effect.

Note: The clock signal for the timeout counter is derived from the sampling point signal of the CAN core. Consequently, the exact timing of the counter's decrements may vary due to the synchronization/resynchronization mechanisms of the CAN core. If the bit rate switching of CAN FD is enabled, the timeout counter will tick at different rates in the arbitration and data fields.

20.5.4 Rx handler

The Rx Handler controls the acceptance filtering, the transfer of received messages to either the Rx buffer or one of the two Rx FIFOs, and manages the put and get indices for the Rx FIFOs.

Note: In the current design, it is not possible to block the reception of debug messages or prevent the toggling of their associated status bits. However, this does not affect the normal usage of the CAN controller.

20.5.4.1 Acceptance filtering

The CAN controller can be configured with two sets of acceptance filters: one for standard identifiers and another for extended identifiers. These filters can be assigned to either the Rx buffer or to Rx FIFO 0/1. During acceptance filtering, each filter list is executed starting from Element 0 until the first matching element is found. The acceptance filtering process stops upon finding the first match, and subsequent filter elements in the list are not evaluated.

The key features are as follows:

- Each filter element can be configured for either acceptance or rejection filtering.
- Each filter element can be individually enabled or disabled.
- Each filter element can be configured as:
  - A filter for one or two dedicated IDs
  - A classic bit mask filter

- A range filter
- Filters are checked sequentially. The execution stops as soon as the first matching filter element is found.

Related configuration registers:

- Standard ID Filter Configuration (SIDFC)
- Extended ID and Mask (XIDAM)
- Extended ID Filter Configuration (XIDFC)
- Global Filter Configuration (GFC)

Depending on the configuration of the filter element (SFEC/EFEC), a match triggers one of the following actions:

- Set the high-priority message interrupt flag IR[HPM]
- Set the high-priority message interrupt flag IR[HPM] and store the received frame in FIFO 0 or FIFO 1
- Store the received frame in FIFO 0 or FIFO 1
- Store the received frame in the Rx Buffer
- Reject the received frame

Acceptance filtering begins after the complete identifier has been received. Once acceptance filtering is complete and a matching Rx Buffer or Rx FIFO is found, the message handler writes the received message data in 32-bit units to the matched Rx Buffer or Rx FIFO. If the CAN protocol controller detects an error (for example, a CRC error), the message is discarded. This has the following consequences for the affected Rx buffer or Rx FIFO:

- Rx Buffer: The new data flag for the matched Rx buffer is not set, but the Rx buffer is (partially) overwritten by the received data. The error type can be checked in PSR[LEC] or PSR[DLEC].
- Rx FIFO: The put index for the matched Rx FIFO is not updated, but the associated Rx FIFO element is (partially) overwritten by the received data. The error type can be checked in PSR[LEC] or PSR[DLEC]. If the matched Rx FIFO operates in overwrite mode, the boundary conditions of the Rx FIFO in this mode must be considered.

Note: When a received message that has passed acceptance filtering is written to one of the two Rx FIFOs or to a Rx Buffer, the unmodified received identifier is stored, regardless of which filters were used. The outcome of the acceptance filtering process depends heavily on the order of the configured filter elements.

Range filter

This filter matches all received frames whose message IDs fall within the range defined by SF1ID/SF2ID or EF1ID/EF2ID.

When range filter works with extended frames, there are two situations:

EFT = 00: The AND operation of received frame’s message ID and the Extended ID and Mask (XIDAM) will be implemented before the range filter is

applied.

EFT = 11: The range filter is applied without using the Extended ID and Mask (XIDAM).

Specific message ID filter

A single filter element can be configured to filter one or two specific message IDs. To filter a single specific message ID, the filter element must be configured such that SF1ID = SF2ID or EF1ID = EF2ID.

Classic bit mask filter

Classic bit mask filter is designed to filter groups of message IDs by masking individual bits of the received message IDs. In classic bit mask filter, SF1ID/EF1ID serves as the message ID filter, while SF2ID/EF2ID serves as the filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter. For example, the value of the received message ID in that bit position is irrelevant for acceptance filtering. Only those bit positions where the mask bit is “1” in the received message IDs are significant for acceptance filtering.

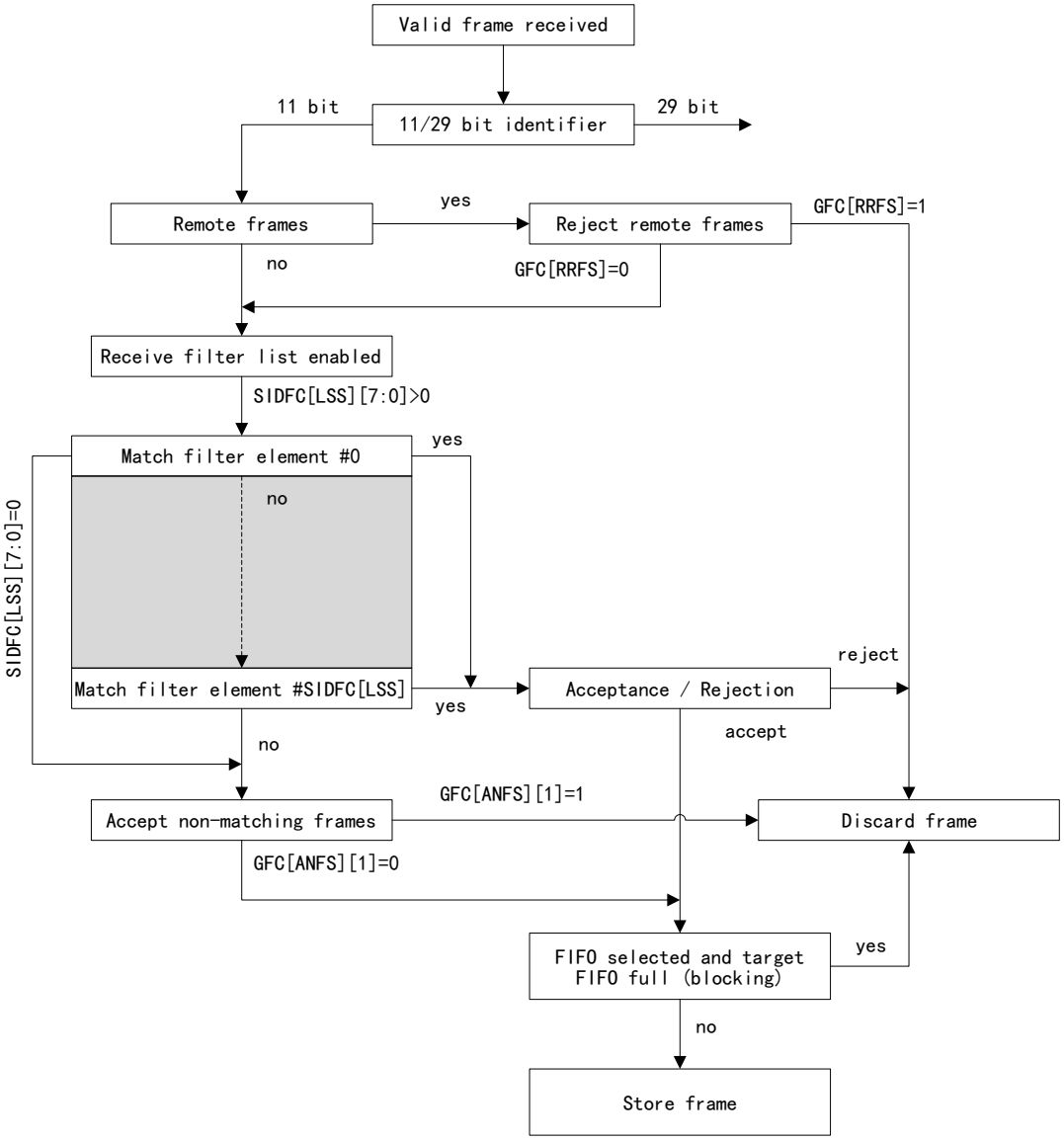
If all mask bits are “1”, a match occurs only when the received message ID is identical to the message ID filter. If all mask bits are “0”, any message ID will match.

Standard message ID filter

The following figure illustrates the filtering flow for a standard message ID (11-bit identifier). The standard message ID filter elements are described in other sections.

Under the control of the Global Filter Configuration (GFC) and Standard ID Filter Configuration (SIDFC) registers, the received frame’s message ID, Remote Transmission Request (RTR) bit, and Identifier Extension (IDE) bit are compared against the configured list of filter elements.

Figure 80 Standard Message ID (11-bit identifier) Filtering Flow Chart



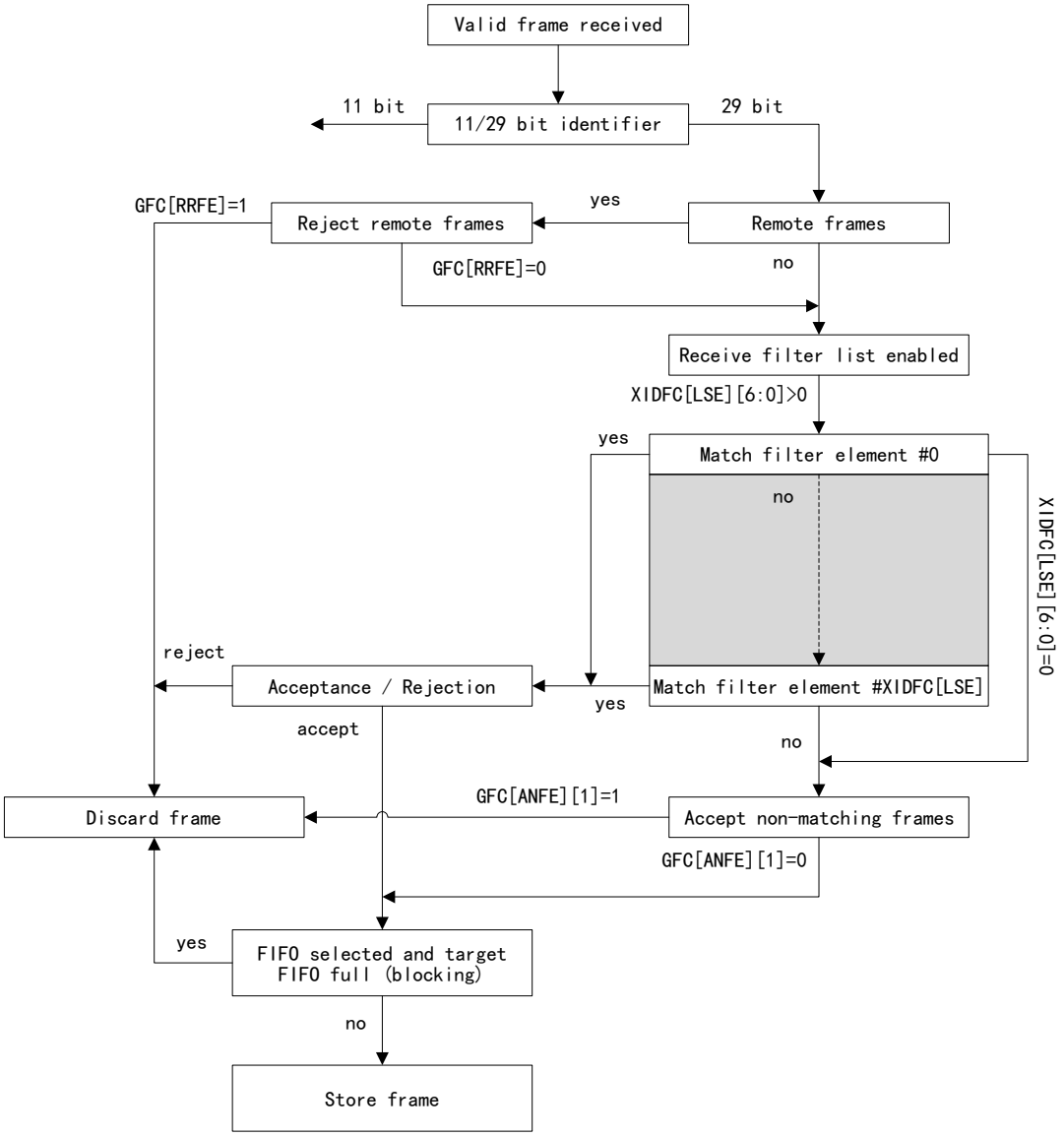
Extension message ID filter

The following figure illustrates the filtering flow for an extension message ID (29-bit identifier). The extension message ID filter elements are described in other sections.

Under the control of the Global Filter Configuration (GFC) and Extension ID Filter Configuration (XIDFC) registers, the received frame's message ID, Remote Transmission Request (RTR) bit, and Identifier Extension (IDE) bit are compared against the configured list of filter elements.

The AND operation of the Extended ID and Mask (XIDAM) and the received identifier will be implemented before the filter list is applied.

Figure 81 Extension Message ID (29-bit identifier) Filtering Flow Chart

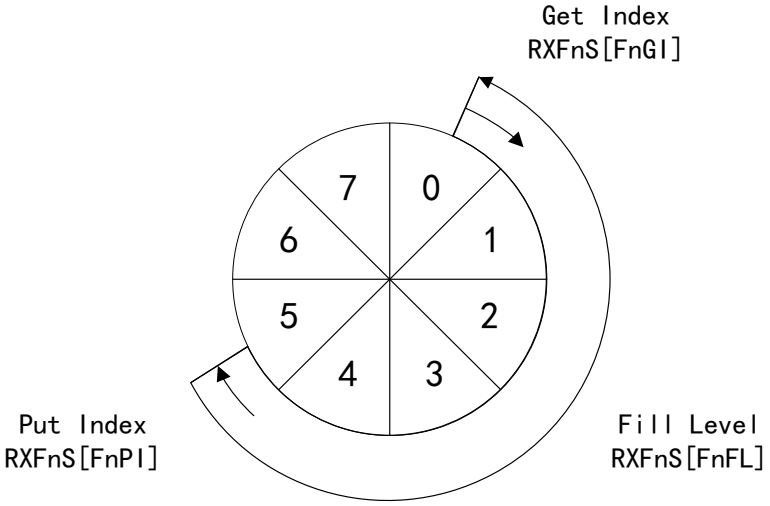


20.5.4.2 Rx FIFO

Both Rx FIFO 0 and Rx FIFO 1 can be individually configured to hold up to 64 elements. The two Rx FIFOs are configured via the RXF0C and RXF1C registers. Received messages that pass the acceptance filtering are transferred to the Rx FIFO based on the configuration of the matching filter element. For a description of the Rx FIFO element structure and the filtering mechanisms available for Rx FIFO 0 and Rx FIFO 1, refer to the relevant sections.

To prevent Rx FIFO overflow, an Rx FIFO watermark can be used. When the fill level of the Rx FIFO reaches the watermark value configured by RXFnC[FnWM], the interrupt flag IR[RFnW] is set. When the number of Rx FIFO's put indexes reaches the get level, RXFnS[FnF] signals that the Rx FIFO is full. Additionally, the interrupt flag IR[RFnF] is set.

Figure 82 Rx FIFO Status



When reading data from Rx FIFO, the Rx FIFO get index RXFnS[FnGI] must be multiplied by the FIFO element size and then added to the corresponding Rx FIFO start address RXFnC[FnSA].

Table 68 Rx FIFO and FIFO Element Size

RXESC[RBDS][2:0] RXESC[FnDS][2:0]	Data Field (bytes)	FIFO Element Size (RAM words)
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

Rx FIFO Blocking Mode

Rx FIFO Blocking Mode is configured by setting RXFnC[FnOM] to 0. This is the default operating mode of the Rx FIFO.

When the condition for a full Rx FIFO is met (RXFnS[FnPI] = RXFnS[FnGI]), no further messages will be written to the corresponding Rx FIFO until at least one message is read, causing the Rx FIFO get index to increment. RXFnS[FnF] = 1 signals that the Rx FIFO is full. Additionally, the interrupt flag IR[RFnF] is set.

If a message is received while the corresponding Rx FIFO is full, the message is discarded, and RXFnS[RFnL] = 1 signals a message loss. Additionally, the interrupt flag IR[RFnL] is set.

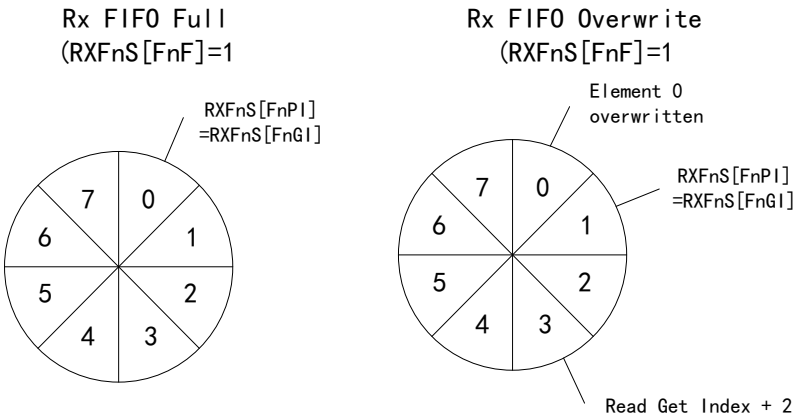
Rx FIFO Overwrite Mode

Rx FIFO Overwrite Mode is configured by setting RXFnC[FnOM] to 1.

When RXFnS[FnF] = 1 and it signals that the Rx FIFO is full (RXFnS[FnPI] = RXFnS[FnGI]), the next message accepted by the FIFO will overwrite the oldest FIFO message. Both the put index and the get index are incremented by 1.

When Rx FIFO operates in Overwrite Mode and signals that it is full, reading the Rx FIFO elements should start from at least the get index + 1. This is necessary because a situation may arise where the CPU is reading data from the message RAM (at the get index) while a received message is simultaneously being written to the message RAM (at the put index). In this case, inconsistent data might be read from the corresponding Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The required offset depends on the speed of the CPU's access to the Rx FIFO. The figure below illustrates a scenario where an offset of 2 relative to the get index is used during FIFO read operations. In this case, the two messages stored in Element 1 and 2 are lost.

Figure 83 Rx FIFO Overflow Processing



After reading data from an Rx FIFO, it must write the element number of the last read element to the Rx FIFO Acknowledge Index RXFnA[FnAI]. This action increments the get index to that element number. If the put index has not yet incremented to this Rx FIFO element, the Rx FIFO full condition will be reset (RXFnS[FnF] = 0).

20.5.4.3 Dedicated Rx buffer

The CAN controller supports up to 64 dedicated Rx buffers. The start address of the dedicated Rx buffer is configured via RXBC[RBSA].

For each Rx buffer, a standard or extension message ID filter element must be configured with SFEC/EFEC = 111 and SFID2/EFID2[10:9] = 00.

After a received message is accepted by a filter element, it is stored into the Rx buffer within the message RAM that is referenced by that filter element. Its

format is identical to that of the Rx FIFO element. Furthermore, the flag IR[DRX] (message stored to a dedicated Rx buffer) in the interrupt register is set.

Table 69 Rx Buffer Filter Configuration Example

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[5:0] EFID2[5:0]	SFID2[10:9] EFID2[10:9]
0	ID Message 1	00 0000	00
1	ID Message 2	00 0001	00
2	ID Message 3	00 0010	00

After the last word of the matching message received is written into the message RAM, the corresponding new data flag in the NDAT1 or NDAT2 register is set. As long as the new data flag is set, the corresponding Rx buffer is locked to prevent newly received matching frames from being updated. The host must reset the new data flag by writing “1” to the corresponding bit position.

When the new data flag of a Rx buffer is set, the message ID filter element referencing that specific Rx buffer will not produce a match, allowing the acceptance filtering process to continue. Subsequent message ID filter elements may cause the received message to be stored in another Rx buffer or the Rx FIFO, or rejected, according to the filter configuration.

Rx buffer handling

- Reading the message from the message RAM
- Resetting the interrupt flag IR[DRX]
- Resetting the new data flags for the processed messages
- Reading the new data register

20.5.5 Tx Handler

The Tx Handler is responsible for handling transmission requests from dedicated Tx buffers, Tx FIFOs, and Tx queues. It controls the transfer, the write and read indices of messages to be transmitted to the CAN core, as well as the Tx Event FIFO. Up to 32 Tx buffers can be configured for message transmission. The transmission CAN mode (classic CAN or CAN FD) for each Tx buffer element can be individually configured. The table below illustrates the possible configurations for frame transmission.

Table 70 Configurations for Frame Transmission

Tx Buffer Element		CCCR		Frame Transmission
FDF	BRS	BRSE	FDOE	
ignored	ignored	ignored	0	Classic CAN
0	ignored	0	1	Classic CAN
1	ignored	0	1	FD without bit rate switching
0	ignored	1	1	Classic CAN

Tx Buffer Element		CCCR		Frame Transmission
FDF	BRS	BRSE	FDOE	
1	0	1	1	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Note: AUTOSAR requires at least three Tx queue buffers and supports transmission cancellation.

When the Tx Buffer Request Pending Register (TXBRP) is updated, the Tx Handler performs a Tx scanning of the message RAM to check for the highest-priority pending Tx request (the Tx buffer with the lowest message ID). TXBRP is updated when a transmission is completed, or after the host requests transmission by writing to TXBAR, or after the host cancels a pending transmission by writing to TXBCR.

When new scan results are available, temporary buffers are updated. These temporary buffers store the preloaded first four RAM words of the two messages with the highest Tx priority. The time required for Tx scanning and preloading depends on the host clock frequency, the number of configured Tx buffers, and the number of CAN modules connected to the message RAM.

Since the Tx scanning of the message RAM takes some time, the following situations may occur:

- Preloading of the temporary buffers is not completed before an ongoing transmission/ reception finishes: The Tx message cannot start transmission at the first opportunity. In this case, another node may start transmission without having to arbitrate with the CAN's transmission message. If this other message has a lower priority, this may be considered external priority inversion.
- Preloading of the temporary buffers is completed before an ongoing transmission/reception finishes: The preloaded Tx message can start transmission at the first opportunity.

20.5.5.1 Transmission pausing

The Transmission Pausing feature is applicable to CAN systems where CAN message identifiers are (permanently) assigned specific values and are not easily changed. These message identifiers might have a higher CAN arbitration priority than other defined messages, whereas in specific applications, their relative arbitration priority should be the opposite. This can lead to a situation where one ECU sends a series of CAN messages, causing CAN messages from another ECU to be delayed because these other messages have a lower CAN arbitration priority.

For example, if CAN ECU-1 enables the Transmission Pausing feature and its application software requests the transmission of four messages, then after the first successful message transmission, it will wait for two CAN bit times of bus idle time before allowing the next requested message transmission to begin. If

other ECUs have messages pending for transmission, these messages will start transmission during the idle time and do not need to arbitrate with ECU-1's next message. After receiving a message, ECU-1 can start its next transmission once the received message has released the CAN bus.

The Transmission Pausing feature is controlled by the CCCR[TXP] bit. If this bit is set to 1, the CAN will pause for the time of two CAN bits after each successful message transmission before starting the next transmission. This allows other CAN nodes in the network to transmit messages, even if their messages have lower-priority identifiers. By default, the Transmission Pausing feature is disabled (CCCR[TXP] = 0).

This feature mitigates burst transmissions from a single node and prevents "erroneous instructions," where the application incorrectly requests an excessive number of transmissions.

20.5.5.2 Dedicated Tx buffer

Dedicated Tx buffers are used for message transmission under the full control of the host CPU. Each dedicated Tx buffer is configured with a specific message ID. When multiple Tx buffers are configured with the same message ID, the Tx buffer with the lowest buffer number implements the transmission first. This can be achieved in one of two ways:

- Tx buffers should be requested in ascending numerical order, starting with the lowest-numbered buffer.
- All Tx buffers configured with the same message ID can be requested simultaneously via a single write access to the TXBAR register.

A transmission is requested by sending an Add Request via the TXBAR[Arn] bit, typically after the message data has been updated. The requested message arbitrates internally with messages from the optional Tx FIFO or Tx queues, and externally with messages on the CAN bus, before being transmitted based on their message IDs.

A dedicated Tx buffer is allocated an element in the message RAM consisting of 32-bit words (see the table below). Consequently, the start address of a dedicated Tx buffer in the message RAM is calculated by multiplying the Tx Buffer Index (0…31) by the element size and adding it to the Tx buffer start address (TXBC[TBSA]).

Table 71 Tx Buffer, FIFO and Queue Element Size

TXESC[TBDS][2:0]	Data Field (bytes)	Element Size (RAM words)
000	8	4
001	12	5
010	16	6
011	20	7

TXESC[TBDS][2:0]	Data Field (bytes)	Element Size (RAM words)
100	24	8
101	32	10
110	48	14
111	64	18

20.5.5.3 Tx FIFO

Tx FIFO operation is configured by programming TXBC[TFQM] to 0. Messages stored in the Tx FIFO are transmitted starting from the message referenced by the read index TXFQS[TFGI]. After each transmission, the read index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO allows the transmission of messages with the same message ID from different Tx buffers in the order they were written into the Tx FIFO. The CAN core calculates the Tx FIFO Free Level TXFQS[TFFL] as the difference between the write index and the read index. It indicates the number of available (free) Tx FIFO elements.

New transmission messages must be written into the Tx FIFO starting from the Tx buffer referenced by the write index TXFQS[TFQPI]. An Add Request increments the write index to the next free Tx FIFO element. When the write index reaches the read index, a Tx FIFO Full condition (TXFQS[TFQF]=1) is signaled. In this case, no further messages should be written to the Tx FIFO until the next message is transmitted and the read index is incremented.

When adding a single message to the Tx FIFO, transmission is requested by writing “1” to the TXBAR bit associated with the Tx buffer referenced by the Tx FIFO’s write index.

When adding multiple (n) messages to the Tx FIFO, they are written into n consecutive Tx buffers starting from the write index. Transmission is then requested via TXBAR. The write index is cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx buffers indicated by the Tx FIFO Free Level.

When a transmission request is cancelled for the Tx buffer referenced by the read index, the read index is incremented to the next Tx buffer with a pending transmission request, and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx buffer, the read index and FIFO free level remain unchanged.

A Tx FIFO element is allocated in the Message RAM with an element size of 32-bit words. Therefore, the start address of the next available (free) Tx FIFO buffer is calculated by multiplying the Tx FIFO/Queue Write Index TXFQS[TFQPI] (0…31) by the element size and adding it to the Tx Buffer Start Address TXBC[TBSA].

20.5.5.4 Tx Queue

Tx queue operation is configured by programming TXBC[TFQM] to 1. Messages stored in the Tx queue are transmitted starting from the message with the lowest Message ID (highest priority). If multiple queue buffers are configured with the same Message ID, the transmission order depends on the buffer numbers where the messages are stored for transmission. Since these buffer numbers depend on the current state of the write index, the transmission order cannot be predicted.

New messages must be written to the Tx buffer referenced by the write index TXFQS[TFQPI]. The write index always points to the lowest-numbered free buffer in the Tx queue. If the Tx queue is full (TXFQS[TFQF]=1), the write index becomes invalid, and no further messages should be written to the Tx queue until at least one requested message has been transmitted or a pending transmission request has been cancelled.

The APP can use the TXBRP register instead of the write index and can place messages into any Tx buffer that does not have a pending transmission request.

A Tx queue buffer is allocated in the Message RAM with an element size of 32-bit words. Therefore, the start address of the next available (free) Tx queue buffer is calculated by multiplying the Tx FIFO/Queue Write Index TXFQS[TFQPI] (0…31) by the element size and adding it to the Tx Buffer Start Address TXBC[TBSA].

20.5.5.5 Mixed Dedicated Tx Buffers / Tx FIFO

In this configuration, the Tx buffer section in the Message RAM is divided into a set of Dedicated Tx Buffers and one Tx FIFO.

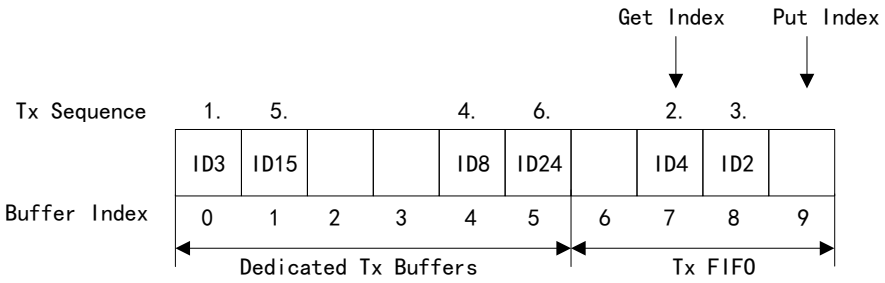
- The number of Dedicated Tx Buffers is configured by TXBC[NDTB].
- The number of Tx Buffers allocated to the Tx FIFO is configured by TXBC[TFQS].

Note: If TXBC[TFQS] is programmed to zero, only Dedicated Tx Buffers are used.

Tx Priority:

- The buffer with the lowest Message ID has the highest priority and will be transmitted next.
- Scans both the Dedicated Tx Buffers and the oldest pending Tx FIFO buffer (referenced by TXFQS[TFGI]) during next transmission.

Figure 84 Example of a Mixed Configuration with Dedicated Transmission Buffers / Transmission FIFO



20.5.5.6 Mixed Dedicated Tx Buffers / Tx Queue

In this configuration, the Tx buffer section in the Message RAM is divided into a set of Dedicated Tx Buffers and one Tx Queue.

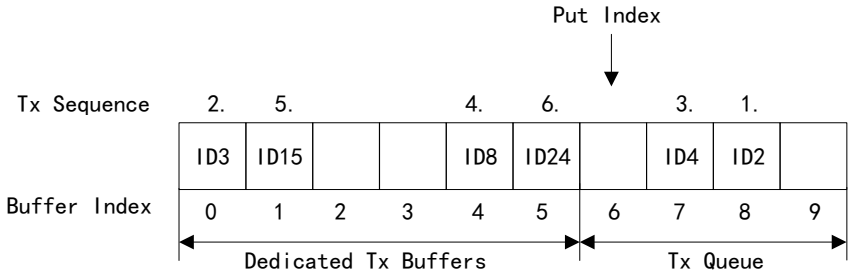
- The number of Dedicated Tx Buffers is configured by TXBC[NDTB].
- The number of Tx Queue Buffers is configured by TXBC[TFQS].

Note: If TXBC[TFQS] is programmed to zero, only Dedicated Tx Buffers are used.

Tx Priority:

- The buffer with the lowest Message ID has the highest priority and will be transmitted next.
- Scan all Tx Buffers with the transmission enabling request during next transmission.

Figure 85 Example of a Mixed Configuration with Dedicated Transmission Buffers / Transmission Queues



20.5.5.7 Transmission cancellation

CAN supports transmission cancellation. This feature is particularly useful for gateway applications and AUTOSAR-based applications.

To cancel a transmission request from the Dedicated Tx Buffer or Tx Queue Buffer, the host must write “1” to the corresponding bit (= Tx Buffer number) of the TXBCR register. A successful cancellation is indicated by setting the corresponding bit in the TXBCF register to “1”.

Transmission cancellation does not apply to Tx FIFO operation. If transmission cancellation is requested for a Tx Buffer while its transmission is already

ongoing, the corresponding TXBRP bit will remain set as long as the transmission is still in progress. If the transmission is successful, the corresponding TXBTO and TXBCF bits will be set. If the transmission fails, the message will not be retransmitted and only the corresponding TXBCF bit will be set.

Note: There is a short time window where, if a pending transmission is cancelled immediately before its transmission is about to start, no transmission will be initiated by this node even if it has other pending messages. This may allow another node to transmit a message whose priority is lower than that of the second message in this node.

20.5.5.8 Tx event handling

To support Tx event handling, the CAN module implements a Tx Event FIFO. After a message is transmitted on the CAN bus, the Message ID and the timestamp are stored in the Tx Event FIFO element. To link the Tx event to the Tx Event FIFO element, the Message Marker from the transmitted Tx buffer is copied into the Tx Event FIFO element.

The configuration of CCCR[WMM] determines whether an 8-bit or 16-bit Message Marker is used. When CCCR[WMM] = 1, the internal timestamp function is disabled.

The Tx Event FIFO can be configured for up to 32 elements.

The purpose of the Tx Event FIFO is to decouple the handling of transmission status information from the handling of the transmitted messages themselves. That is, the Tx buffers only hold the messages to be transmitted, while the transmission status information is stored separately in the Tx Event FIFO. This offers some advantages, particularly during dynamic management of transmission queues, as the Tx buffers can be immediately reused for a new message after a successful transmission. There is no need to preserve the transmission status information of the Tx buffers before overwriting them.

When IR[TEFF] signals that the Tx Event FIFO is full, no new elements are written to it until at least one element is read and the Tx Event FIFO read index is incremented. If a Tx event occurs while the Tx Event FIFO is full, the event is discarded and the interrupt flag IR[TEFL] is set.

To prevent Tx Event FIFO overflow, a Tx Event FIFO watermark can be used. The interrupt flag IR[TEFW] is set when the fill level of the Tx Event FIFO reaches the Tx Event FIFO watermark configured by TXEFC[EFWM].

When reading data from the Tx Event FIFO, twice the Tx Event FIFO read index TXEFS[EFGI] must be added to the Tx Event FIFO Start Address TXEFC[EFSA].

20.5.6 FIFO Acknowledge Handling

The read indices for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO are

controlled by writing to the corresponding FIFO Acknowledge Index. Writing to the FIFO Acknowledge Index sets the FIFO read index to the Acknowledge Index value plus one, thereby updating the FIFO fill level. There are two primary use cases:

When reading only a single element (the element pointed to by the read index) from the FIFO, write this read index value to the FIFO Acknowledge Index.

When reading a sequence of elements from the FIFO, write the FIFO Acknowledge Index only once at the end of the read sequence (using the index of the last read element) to update the FIFO read index.

Since the CPU has free access to the CAN's Message RAM, special care must be taken when reading FIFO elements in an arbitrary order (disregarding the read index). This might be useful, for example, when reading a high-priority message from one of the two Rx FIFOs. In such a case, the FIFO's Acknowledge Index should not be written, as this would set the read index to an incorrect position and also incorrectly change the FIFO fill level. Consequently, some older FIFO elements would be lost.

Note: The application must ensure that the value written to the FIFO Acknowledge Index is valid. The CAN module does not perform checks for erroneous values.

20.5.7 Message RAM

To store received (Rx)/transmitted (Tx) messages and filter configurations, the CAN module must be connected to a 2 KB single-port SRAM, referred to as the Message RAM.

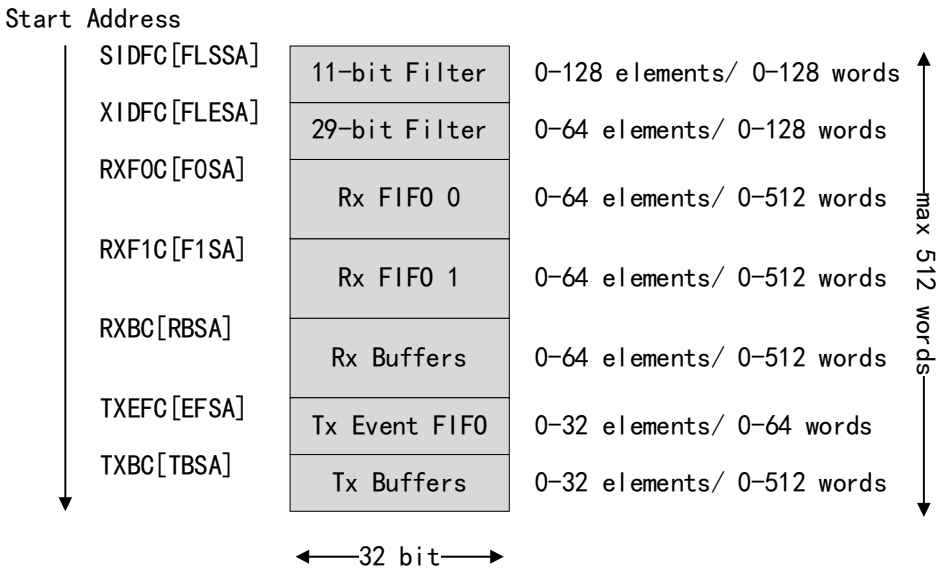
Note: The Message RAM supports ECC (Error Correcting Code) function. It is recommended to initialize the Message RAM after hardware reset. For example, by writing 0x00000000 to each word of the Message RAM to generate valid check bits. This avoids falsely triggering ECC interrupts when reading data from uninitialized sections of the Message RAM, in case the ECC check feature is enabled.

20.5.7.1 Message RAM configuration

The Message RAM has a width of 32 bits. The CAN module can be configured to allocate up to 512 words in the Message RAM. It is not necessary to configure every section listed in the figure below, and there are no restrictions on the order in which the sections are arranged.

In CAN FD mode, the required Message RAM size depends heavily on the element sizes configured via RXESC[F0DS], RXESC[F1DS], RXESC[RBDS], and TXESC[TBDS] for the Receive FIFO 0 (Rx FIFO 0), Receive FIFO 1 (Rx FIFO 1), Receive Buffers (Rx Buffers), and Transmit Buffers (Tx Buffers), respectively.

Figure 86 Message RAM Configuration



When the CAN module addresses the Message RAM, it accesses in 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses. This means that only bits [15:2] of the address are evaluated, and the two least significant bits are ignored.

Note: The CAN module does not check for errors in the Message RAM configuration. Particular care must be taken when configuring the start addresses of different sections and the number of elements per section to avoid data corruption or loss.

20.5.7.2 Rx Buffers and FIFO Elements

Up to 64 Receive Buffers (Rx Buffers) and two Receive FIFOs can be configured in the Message RAM. Each Receive FIFO can be configured to store up to 64 received messages. The table below shows the structure of a Receive Buffer / FIFO element. The element size can be configured via the RXESC register to store CAN FD messages with data fields up to 64 bytes.

Table 72 Rx Buffers and FIFO Elements

Name	31:24			23:16				15:8	7:0
R0	ESI	XTD	RTR	ID[28:0]					
R1	ANMF	FIDX[6:0]		Reserved	FDF	BRS	DLC[3:0]	RXTS[15:0]	
R2	DB3[7:0]			DB2[7:0]				DB1[7:0]	DB0[7:0]
R3	DB7[7:0]			DB6[7:0]				DB5[7:0]	DB4[7:0]
…	…			…				…	…
Rn	DBm[7:0]			DBm-1[7:0]				DBm-2[7:0]	DBm-3[7:0]

R0 Register Bit Description

Bit 31 ESI (Error State Indicator):

This bit indicates the error state of the transmitting node.

0: Indicates that the transmitting node is in the error active state, meaning that this node has not accumulated a critical number of errors and can participate in bus communication normally.

1: Indicates the transmitting node is in the error passive state, meaning that this node has accumulated a significant number of errors and its participation in bus communication is restricted.

Bit 30 XTD (Extended Identifier):

This bit informs the host whether the received frame uses a standard or extended identifier.
0: Indicates that the received frame uses an 11-bit standard identifier which is commonly used in CAN bus communication.

1: Indicates that the received frame uses a 29-bit extended identifier which provides a larger address space for more complex networks or a higher number of nodes.

Bit 29 RTR (Remote Transmission Request):

This bit informs the host whether the received frame is a data frame or a remote frame.

0: Indicates that the received frame is a data frame which contains actual data payload.

1: Indicates that the received frame is a remote frame which does not contain data but requests another node to transmit a data frame with the same identifier.

Note: Remote frames do not exist in the CAN FD format. In a CAN FD frame (FDF=1), a dominant RRS (Remote Request Substitute) bit replaces the RTR bit.

Bits 28:0 ID[28:0] (Identifier):

The type of identifier (standard or extended) is determined by the XTD bit. If XTD = 0 (Standard Identifier), the Standard Identifier is stored in ID[28:18]. If XTD = 1, then use Extended Identifier.

R1 Register Bit Description

Bit 31 ANMF (Accepted non-matching frame):

Acceptance of non-matching frames can be enabled via GFC[ANFS] and GFC[ANFE].

0: Indicates that the received frame matched a filter index FIDX.

1: Indicates that the received frame did not match any receive filter element.

Bits 30-24 FIDX[6:0] (Filter Index):

This value, in the range of 0-127, represents the index of the matching Receive Accept Filter element (this value is invalid if ANMF=1). The valid range is 0 to SIDFC[LSS]-1 (for Standard Identifiers) or 0 to XIDFC[LSE]-1 (for Extended Identifiers).

Bit 21 FDF (FD Format):

This bit indicates the frame format.

0: Legacy CAN frame format.

1: CAN FD frame format (uses new DLC code and CRC).

Bit 20 BRS (Bit Rate Switch):

This bit indicates whether the received frame performed bit rate switching.

0: Indicates that the received frame did not use bit rate switching;

1: Indicates that the received frame did use bit rate switching.

Bits 19-16 DLC[3:0] (Data Length Code):

In both CAN and CAN FD, a value of 0-8 indicates the received frame has 0-8 data bytes.

In CAN, a value of 9-15 indicates the received frame has 8 data bytes.

In CAN FD, a value of 9-15 indicates the received frame has 12/16/20/24/32/48/64 data bytes respectively.

Bits 15-0 RXTS[15:0] (Receive Timestamp):

This field captures the value of the Timestamp Counter when the frame's reception started. Its resolution depends on the configuration of the Timestamp Counter Prescaler TSCC[TCP].

R2-Rn Register Bit Description

These registers are used to store the data bytes of CAN messages.

For examples: Bits 31-24 of Register R2 store Data Byte 3 (DB3[7:0])

Bits 23-16 of Register R2 store Data Byte 2 (DB2[7:0])… and so on.

Note: Depending on the configured element size (RXESC), 2 to 16 words of 32 bits (Rn=2…17) are used to store the data field of CAN messages.

20.5.7.3 Tx buffer elements

The Transmit Buffer section can be configured to Dedicated Transmit Buffers as well as a Transmit FIFO (Tx FIFO) / Transmit Queue. If the Transmit Buffer section is shared by Dedicated Transmit Buffers and a Tx FIFO / Tx Queue,

then the Dedicated Transmit Buffers start at the beginning of the Transmit Buffer section, followed by the buffers allocated to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between Dedicated Transmit Buffers and the Tx FIFO / Tx Queue by evaluating the transmit buffer configuration registers TXBC[TFQS] and TXBC[NDTB]. The element size can be configured via the TXESC register to store CAN FD messages with data fields up to 64 bytes in length.

Table 73 Tx Buffer Elements

Name	31:24			23:16					15:8	7:0
T0	ESI	XTD	RTR	ID[28:0]						
T1	MM[7:0]			EFC	Reserved	FDF	BRS	DLC[3:0]	MM[15:8]	Reserved
T2	DB3[7:0]			DB2[7:0]					DB1[7:0]	DB0[7:0]
T3	DB7[7:0]			DB6[7:0]					DB5[7:0]	DB4[7:0]
…	…			…					…	…
Tn	DBm[7:0]			DBm-1[7:0]					DBm-2[7:0]	DBm-3[7:0]

T0 Register Bit Description

Bit 31 ESI (Error State Indicator):

0: In CAN FD format, the ESI bit depends solely on the Error Passive Flag.

1: The transmitted ESI bit in CAN FD format is set to recessive.

Note: The “OR” operation between ESI bit in the Transmit Buffer and the Error Passive Flag is implemented to determine the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, a node in the error active state may optionally transmit a recessive ESI bit, but a node in the error passive state will always transmit a recessive ESI bit.

Bit 30 XTD (Extended Identifier):

0: 11-bit standard identifier.

1: 29-bit extended identifier.

Bit 29 RTR (Remote Transmission Request):

0: Transmit a data frame.

1: Transmit a remote frame.

Note: When RTR=1, even if CCCR[FDOE] enables transmission in CAN FD format, the CAN module will transmit a remote frame according to the ISO 11898-1:2015.

Bits 28:0 ID[28:0] (Identifier):

The type of identifier (standard or extended) is determined by the XTD bit. Standard identifiers must be written into ID[28:18].

T1 Register Bit Description

Bits 31-24 MM[7:0] (Message Marker):

Written by the CPU during Tx Buffer configuration. It is copied into the Tx Event FIFO element to identify the status of the transmitted message.

Bit 23 EFC (Event FIFO Control):

0: Do not store a Tx event.

1: Store a Tx event.

Bit 21 FDF (FD Format):

0: Transmit the frame in classic CAN format.

1: Transmit the frame in CAN FD format.

Bit 20 BRS (Bit Rate Switch):

0: The CAN FD frame is transmitted without bit rate switching.

1: The CAN FD frame is transmitted with bit rate switching.

Note: The ESI, FDF, and BRS bits are only evaluated when CAN FD operation is enabled (CCCR[FDOE]=1). Additionally, the BRS bit is only evaluated when CCCR[BRSE]=1.

Bits 19-16 DLC[3:0] (Data Length Code):

0-8: In both CAN and CAN FD, the transmitted frame has 0-8 data bytes.

9-15: In CAN, the transmitted frame has 8 data bytes; in CAN FD, the transmitted frame has 12/16/20/24/32/48/64 data bytes.

Bits 15–8 MM[15:8] (Message Marker):

The high bytes of the Message Marker, are written by the CPU during Tx Buffer configuration. It is copied into the Tx Event FIFO element to identify the status of the transmitted message. Only available when CCCR[WMM]=1.

T2-Tn Register Bit Description

Bits 31-24 of T2 correspond to DB3[7:0] (Data Byte 3)

Bits 23-16 correspond to DB2[7:0] (Data Byte 2)

Bits 15-8 correspond to DB1[7:0] (Data Byte 1)

Bits 7-0 correspond to DB0[7:0] (Data Byte 0)

This pattern continues in T3 register and subsequent registers, used for storing the data bytes of the CAN message.

Note: Depending on the configured element size (TXESC), 2-16 words of 32 bits (Tn=2…17) are used to store the data field of CAN messages.

20.5.7.4 Tx Event FIFO elements

Each element stores information about a transmitted message. The host CPU can retrieve this information in the order the messages were sent by reading the Tx Event FIFO. The status information on the Tx Event FIFO can be obtained from the TXEFS register.

E1A: When CCCR[WMM] = 0, E1A[TXTS][15:0] holds the 16-bit timestamp generated by the CAN internal timestamp logic.

E1B: When the 16-bit Message Marker is enabled (CCCR[WMM] = 1), E1B[MM][15:8] holds the upper 8 bits of the wide Message Marker.

Table 74 Tx Event FIFO Elements

Name	31:24			23:16				15:8	7:0
E0	ESI	XTD	RTR	ID[28:0]					
E1A	MM[7:0]			ET[1:0]	FDF	BRS	DLC[3:0]	TXTS[15:0]	
E1B	MM[7:0]			ET[1:0]	FDF	BRS	DLC[3:0]	MM[15:8]	Reserved

E0 Register Bit Description

Bit 31 ESI (Error State Indicator):

0: The transmitting node is in the error active state.

1: The transmitting node is in the error passive state.

Bit 30 XTD (Extended Identifier):

0: 11-bit standard identifier.

1: 29-bit extended identifier.

Bit 29 RTR (Remote Transmission Request):

0: A data frame was transmitted.

1: A remote frame was transmitted.

Bits 28:0 ID[28:0] (Identifier):

The type of identifier (standard or extended) is determined by the XTD bit. Standard identifiers must be stored into ID[28:18].

E1A/B Register Common Bit Description

Bits 31-24 MM[7:0] (Message Marker):

Copied from the Transmit Buffer into the Tx Event FIFO element to identify the status of the transmitted message.

Bits 23-22 ET[1:0] (Event Type):

00: Reserved.

01: Transmission Event.

10: Transmission occurred despite cancellation (Transmissions in DAR mode always set this status).

11: Reserved.

Bit 21 FDF (FD Format):

0: Legacy CAN frame format.

1: CAN FD frame format (uses new Data Length Code (DLC) encoding and Cyclic Redundancy Check (CRC)).

Bit 20 BRS (Bit Rate Switch):

0: The frame was transmitted without bit rate switching.

1: The frame was transmitted with bit rate switching.

Bits 19-16 DLC[3:0] (Data Length Code):

0-8: In both CAN and CAN FD, the transmitted frame has 0-8 data bytes.

9-15: In CAN, the transmitted frame has 8 data bytes; in CAN FD, the transmitted frame has 12/16/20/24/32/48/64 data bytes.

E1A, Bits 15-0 TXTS[15:0] (Transmit Timestamp):

Captures the value of the Timestamp Counter when the frame's transmission started. Its resolution depends on the configuration of the Timestamp Counter Prescaler TSCC[TCP].

E1B, Bits 15-8 MM[15:8] (Message Marker):

The high byte of the Message Marker, written by the CPU during Tx Buffer configuration. It is copied into the Tx Event FIFO element to identify the status of the transmitted message.

20.5.7.5 Standard message ID filter element

The 11-bit IDs can be configured with up to 128 filter elements. When accessing a standard message ID filter element, its address is the start address SIDFC[FLSSA] of Message Filter List plus the filter element index (ranging from 0 to 127).

Table 75 Standard Message ID Filter Elements

Name	31:24		23:16	15:8	7:0
S0	SFT[1:0]	SFEC[2:0]	SFID1[10:0]	Reserved	SFID2[10:0]

Bits 31-30 SFT[1:0] (Standard Filter Type):

00: Range filter from SFID1 to SFID2 (requires SFID2≥SFID1).

01: Dual ID filter for either SFID1 or SFID2.

10: Classic filter: SFID1 serves as the filter, SFID2 serves as the mask.

11: Filter element disabled.

Note: When SFT = “11”, the filtered elements are disabled, and acceptance filtering continues (behaving the same as SFEC = “000”).

Bits 29-27 SFEC[2:0] (Standard Filter Element Configuration):

All enabled filter elements are used for acceptance filtering of frames with 11-bit IDs. Acceptance filtering stops at the first matching enabled filter element or at the end of the filter list. If SFEC is “100”, “101”, or “110”, a match sets the interrupt flag IR[HPM] and, if the interrupt feature is enabled, an interrupt will be generated. In this case, the HPMS register is updated with the priority match status.

000: Disable filter element.

001: If filter matches, the register will store message in Rx FIFO 0.

010: If filter matches, the register will store message in Rx FIFO 1.

011: If filter matches, the register will reject this ID (not suitable for synchronous messages).

100: If filter matches, the register will set priority (not suitable for synchronous messages, no storage).

101: If filter matches, the register will set priority and store message in FIFO 0.

110: If filter matches, the register will set priority and store message in FIFO 1.

111: The message is store into a Receive Buffer or as a Debug message (the SFT[1:0] configuration is ignored in this case).

Bits 26-16 SFID1[10:0] (Standard Filter ID 1):

This is the first ID for the standard ID filter element. When filtering for a Receive Buffer, synchronous message, or debug message, this field defines the ID of the message to be stored. The received identifier must match exactly; no masking mechanism is used.

Bits 10-0 SFID2[10:0] (Standard Filter ID 2):

The meaning of this bit field depends on the SFEC configuration:

When SFEC has a value between “001” and “110”, it is the second ID for the standard ID filter element.

When SFEC = “111”, it is used for filtering for a Receive Buffer or Debug message:

SFID2[10:9] determine whether the received message is stored into a receive buffer or handled as Message A, B, or C in a debug message sequence:

00: Store message into the receive buffer.

01: Debug Message A.

10: Debug Message B.

11: Debug Message C.

Note: CAN module does not support debug messages.

SFID2[8:6]: Reserved.

SFID2[5:0] define the offset, relative to the Receive Buffer Start Address RXBC[RBSA], used for storing the matching messages.

20.5.7.6 Extended message ID filter element

The 29-bit IDs can be configured with up to 64 filter elements. The address for accessing an extended message ID filter element is calculated as the extended filter list start address XIDFC[FLESA] plus double the filter element index (range: 0 to 63).

Table 76 Extended Message ID Filter Elements

Name	31:24		23:16	15:8	7:0
F0	EFEC[2:0]		EFID1[28:0]		
F1	EFT[1:0]	Reserved	EFID2[28:0]		

F0 register

Bits 31-29 EFEC[2:0] (Extended Filter Element Configuration):

All enabled filter elements are used for acceptance filtering of frames with 29-bit IDs. Acceptance filtering stops at the first matching enabled filter element or at the end of the filter list. If EFEC is “100”, “101”, or “110”, a match sets the interrupt flag IR[HPM] and, if the interrupt feature is enabled, interrupts will be generated. In this case, the HPMS register is updated with the priority match status.

000: Disable filter element.

001: If filter matches, the message will be stored in Rx FIFO 0.

010: If filter matches, the message will be stored in Rx FIFO 1.

011: If filter matches, the ID will be rejected (not suitable for synchronous messages).

100: If filter matches, the priority will be set (not suitable for synchronous messages, no storage).

101: If filter matches, the priority is set and the message is stored in FIFO 0.

110: If filter matches, the priority is set and the message is stored in FIFO 1.

111: The message is stored into a Receive Buffer or as a Debug message (the EFT[1:0] configuration is ignored in this case).

Bits 28-0 EFID1[28:0] (Extended Filter ID 1):

This is the first ID for the extended ID filter element. When filtering for a Receive Buffer, synchronous message, or debug message, this field defines the ID of the extended message to be stored. The received identifier must match exactly; only XIDAM masking mechanism is used.

F1 register

Bits 31-30 EFT[1:0] (Extended Filter Type):

00: Range filter from EFID1 to EFID2 (requires EFID2 ≥ EFID1).

01: Dual ID filter for either EFID1 or EFID2.

10: Classic filter: EFID1 serves as the filter, EFID2 serves as the mask.

11: Range filter from EFID1 to EFID2 (requires EFID2 ≥ EFID1), XIDAM mask is not applied.

Bits 28-0 EFID2[28:0] (Extended Filter ID 2):

The meaning of this bit field depends on the EFEC configuration:

When EFEC has a value between “001” and “110”, it is the second ID for the extended ID filter element.

When EFEC = “111”, it is used for filtering for Receive Buffer or Debug message:

EFID2[10:9] determine whether the received message is stored into a receive buffer or handled as Message A, B, or C in a debug message sequence:

00: Store message into the receive buffer.

01: Debug Message A.

10: Debug Message B.

11: Debug Message C.

Note: CAN module does not support debug messages.

EFID2[8:6]: Reserved.

EFID2[5:0] define the offset, relative to the Receive Buffer Start Address RXBC[RBSA], used for storing the matching messages.

20.6 Register Address Mapping

Registers are divided into:

- CAN registers, with a base address of 0x4000 6400
- CAN_ERM registers, with a base address of 0x4000 5800

Table 77 CAN Register Address Mapping

Register name	Description	Offset address
CREL	Core Release Register	0x00
ENDN	Endianness Register	0x04
-	Reserved	0x08
DBTP	Data Bit Timing and Prescaler Register	0x0C
TEST	Test Register	0x10
RWD	RAM Watchdog	0x14
CCCR	CC Control Register	0x18
NBTP	Nominal Bit Timing and Prescaler Register	0x1C
TSCC	Timestamp Counter Configuration Register	0x20
TSCV	Timestamp Counter Value Register	0x24
TOCC	Timeout Counter Configuration Register	0x28
TOCV	Timeout Counter Value Register	0x2C
-	Reserved	0x30-0x3C
ECR	Error Counter Register	0x40
PSR	Protocol Status Register	0x44
TDCR	Transmitter Delay Compensation Register	0x48
-	Reserved	0x4C
IR	Interrupt register	0x50
IE	Interrupt enable	0x54
ILS	Interrupt Line Selection Register	0x58
ILE	Interrupt Line Enable Register	0x5C
-	Reserved	0x60-0x7C
GFC	Global Filter Configuration Register	0x80
SIDFC	Standard ID Filter Configuration Register	0x84
XIDFC	Extended ID Filter Configuration Register	0x88
-	Reserved	0x8C
XIDAM	Extended ID and Mask Register	0x90

Register name	Description	Offset address
HPMS	High Priority Message Status Register	0x94
NDAT1	New Data 1	0x98
NDAT2	New Data 2	0x9C
RXF0C	Rx FIFO 0 Configuration	0xA0
RXF0S	Rx FIFO 0 Status	0xA4
RXF0A	Rx FIFO 0 Acknowledge	0xA8
RXBC	Rx Buffer Configuration	0xAC
RXF1C	Rx FIFO 1 Configuration	0xB0
RXF1S	Rx FIFO 1 Status	0xB4
RXF1A	Rx FIFO 1 Acknowledge	0xB8
RXESC	Rx Buffer/ FIFO Element Size Configuration	0xBC
TXBC	Tx Buffer Configuration	0xC0
TXFQS	Tx FIFO/ Queue Status	0xC4
TXESC	Tx Buffer Element Size Configuration	0xC8
TXBRP	Tx Buffer Request Pending	0xCC
TXBAR	Tx Buffer Add Request	0xD0
TXBCR	Tx Buffer Cancel Request	0xD4
TXBTO	Tx Buffer Transmission Occurred	0xD8
TXBCF	Tx Buffer Cancellation Finished	0xDC
TXBTIE	Tx Buffer Transmission Interrupt Enable	0xE0
TXBCIE	Tx Buffer Cancellation Interrupt Enable	0xE4
-	Reserved	0xE8-0xEC
TXEFC	Tx Event FIFO Configuration	0xF0
TXEFS	Tx Event FIFO Status	0xF4
TXEFA	Tx Event FIFO Acknowledge	0xF8
-	Reserved	0x0FC-0x1FC

Table 78 CAN_ERM Register Address Mapping

Register name	Description	Offset address
ERM_CTL	ERM and Low-Power Control Register	0x00
ERM_STS	ERM and Low-Power Status Register	0x04
ERM_ECCLOG	ERM error address register	0x08

20.7 Register Functional Description

The CAN register only supports 32-bit access.

20.7.1 Core Release Register (CREL)

Offset address: 0x000

Reset value: 0x3315 0415

Field	Name	R/W	Description
7:0	DAY	R	Timestamp Day A number, BCD code
15:8	MON	R	Timestamp Month A number, BCD code
19:16	YEAR	R	Timestamp Year A number, BCD code
23:20	SUBSTEP	R	Core Release Sub-step A number, BCD code
27:24	STEP	R	Core Release Step A number, BCD code
31:28	REL	R	Core Version A number, BCD code

20.7.2 Endianness Register (ENDN)

Offset address: 0x0004

Reset value: 0x8765 4321

Field	Name	R/W	Description
31:0	ETV	R	Endianness Test Value Test value: 0x8765 4321

20.7.3 Data Bit Timing and Prescaler Register (DBTP)

Offset address: 0x00C

Reset value: 0x0000 0A33

This register is writable only when both the CCCR[CCE] and CCCR[INIT] bits are set. The CAN bit time can be programmed within a range of 4 to 49 time quanta (tq). The CAN time quantum can be programmed within a range of 1 to 32 can_cclk cycles. Tq = (DBRP + 1) mtq.

Therefore, the length of the bit time is (programmed value) [DTSEG1 + DTSEG2 + 3] tq, or (functional value) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq. The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sampling point.

Field	Name	R/W	Description
3:0	DSJW	R/W	Data (Re)Synchronization Jump Width 0x00 – 0x0F: Valid values are 0 to 15. The hardware interprets this value as the programmed value plus one.
7:4	DTSEG2	R/W	Data Time Period after Sampling Point

Field	Name	R/W	Description
			0x00 – 0x0F: Valid values are 0 to 15. The hardware interprets this value as the programmed value plus one.
12:8	DTSEG1	R/W	Data Time Period before Sampling Point 0x00 – 0x1F: Valid values are 0 to 31. The hardware interprets this value as the programmed value plus one.
15:13	Reserved		
20:16	DBRP	R/W	Data Bit Rate Prescaler 0x00 – 0x1F: This value is used to divide the oscillator frequency to generate the bit time quantum (tq). The bit time is composed of multiples of this quantum. The valid range for the bit timing prescaler is 0 to 31. When TDC = 1, the range is restricted to 0 or 1. The hardware interprets this value as the programmed value plus one.
22:21	Reserved		
23	TDC	R/W	Transmitter Delay Compensation (TDC) 0: Transmitter delay compensation disabled. 1: Transmitter delay compensation enabled.
31:24	Reserved		

Note: With a CAN communication clock (can_cclk) of 8 MHz, the reset value 0x0000 0A33 configures the CAN for a data phase bit rate of 500 kbit/s. The bit rate configured for the CAN FD data node via DBTP must be greater than or equal to the bit rate configured for the arbitration phase via NBTP.

20.7.4 Test Register (TEST)

Offset address: 0x10

Reset value: 0x0000 0000

Write access to the test registers is only enabled after setting the CCCR[TEST] bit to 1. When the CCCR[TEST] bit is cleared, all test register functions are reset to their default values.

Loopback mode and software control of the can_tx pin are hardware test modes. Programming the Tx control field to a value other than “00” may interfere with message transmission on the CAN bus.

Field	Name	R/W	Description
3:0	Reserved		
4	LBCK	R/W	Loopback Mode 0: Reset value, Loopback mode disabled; 1: Loopback mode enabled.
6:5	TX	R/W	Transmit Pin Control (TX) 00: Reset value; the can_tx pin is controlled by the CAN core and is updated at the end of the CAN bit timing. 01: The sample point can be monitored on the can_tx pin. 10: Drive dominant (0) level on the can_tx pin. 11: Drive recessive (1) level on the can_tx pin.
7	RX	R	Receive Pin Monitor the actual value of can_rx pin 0: CAN bus is dominant (can_rx = 0)

Field	Name	R/W	Description
			1: CAN bus is recessive (can_rx = 1)
31:8	Reserved		

20.7.5 RAM Watchdog (RWD)

Offset address: 0x14

Reset value: 0x0000 0000

RAM watchdog monitors the ready output of the message RAM. When accessing the message RAM via the CAN message interface, the message RAM watchdog counter starts with the value configured in RWD[WDC]. When the message RAM indicates that the access has been successfully completed by activating its ready output signal, the counter reloads the value from RWD[WDC]. If the message RAM does not respond before the counter decrements to zero, the counter stops counting and the interrupt flag IR[WDI] is set. The RAM watchdog counter is clocked by the host clock (can_pclk).

Field	Name	R/W	Description
7:0	WDC	R/W	Watchdog Configuration Start value of Message RAM Watchdog Counter. When this value is “00”, the counter is disabled.
15:8	WDV	R	Watchdog Value Actual count value of Message RAM Watchdog
31:16	Reserved		

20.7.6 CC control register (CCCR)

Offset address: 0x18

Reset value: 0x0000 0001

Refer to the “Software Initialization” in this section for details.

Field	Name	R/W	Description
0	INIT	R/W	Initialization 0: Normal running; 1: Initialization start. Note: Due to the synchronization mechanism between the two clock domains, there may be a delay before the value written to INIT can be read back. Therefore, the programmer must ensure that the previously written value to INIT has been accepted by reading INIT before setting a new value to INIT.
1	CCE	R/W	Configuration Change Enable 0: The CPU has no write access to protected configuration registers; 1: The CPU has write access to protected configuration registers (when CCCR[INIT] = 1).
2	ASM	R/W	Restricted Operation Mode The host can set the ASM bit only when both CCE and INIT are set to “1”. This bit can be reset by the host at any time. 0: Normal CAN operation; 1: Restricted operation mode activated.

Field	Name	R/W	Description
3	CSA	R	Clock Stop Acknowledge 0: No clock stop acknowledge; 1: CAN can set this by stopping can_pclk and can_cclk.
4	CSR	R/W	Clock Stop Request 0: No clock stop request; 1: Clock stop requested. When there is a clock stop request, after all pending transmission requests are completed and the CAN bus becomes idle, first INIT is set, and then CSA is set.
5	MON	R/W	Bus Monitoring Mode The host can set the MON bit only when both CCE and INIT are set to “1”. This bit can be reset by the host at any time. 0: Bus Monitoring Mode disabled 1: Bus Monitoring Mode enabled
6	DAR	R/W	Disable Automatic Retransmission 0: Enable automatic retransmission of unsuccessfully transmitted messages; 1: Disable automatic retransmission.
7	TEST	R/W	Enable Test Mode 0: Normal operation, TEST register remains at its reset value; 1: Test mode, write access to the TEST register is enabled.
8	FDOE	R/W	Enable FD Operation 0: Disable 1: Enable
9	BRSE	R/W	Bit Rate Switch Enable 0: Disable transmission bit rate switching; 1: Enable transmission bit rate switching. Note: When CAN FD operation is disabled (FDOE = 0), BRSE is not considered.
10	Reserved		
11	WMM	R/W	Wide Message Marker (WMM) Enables the 16-bit wide message marker. When the 16-bit wide message marker is used (WMM=1), the 16-bit internal timestamp function of the Tx Event FIFO is disabled. 0: Use 8-bit message marker 1: Use 16-bit message marker, replacing the 16-bit timestamp in the Tx Event FIFO
12	PXHD	R/W	Protocol Exception Handling Disable 0: Enable protocol exception handling 1: Disable protocol exception handling Note: When protocol exception handling is disabled, the CAN will transmit an error frame upon detecting a protocol exception condition.
13	EFBI	R/W	Edge Filtering during Bus Integration 0: Disable edge filtering 1: Two consecutive dominant tq are required to detect a hard synchronization edge

Field	Name	R/W	Description
14	TXP	R/W	Transmit Pause If this bit is set, the CAN pauses for two CAN bits before starting the next transmission after successfully transmitting a frame. 0: Disable 1: Enable
15	NISO	R/W	Non ISO Operation If this bit is set, the CAN uses the CAN FD frame format specification V1.0 defined by Bosch CAN FD. 0: CAN FD frame format, compliant with ISO 11898-1:2015 1: CAN FD frame format, compliant with Bosch CAN FD specification V1.0
31:16	Reserved		

20.7.7 Nominal Bit Timing and Prescaler Register (NBTP)

Offset address: 0x1C

Reset value: 0x0600 0A03

This register is writable only after setting CCCR[CCE] and CCCR[INIT]. The CAN bit time can be programmed in the range of 4 to 385 time quanta (tq). The CAN time quantum can be programmed in the range of 1 to 512 can_cclk cycles. Tq = (NBRP + 1) mtq.

NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2. Therefore, the bit time length is (Programmed value): [NTSEG1 + NTSEG2 + 3] tq or (Functional value): [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq.

An Information Processing Time (IPT) is zero, meaning that the next bit data is available at the first clock edge after the sampling point.

Field	Name	R/W	Description
6:0	NTSEG2	R/W	Nominal Time Period after Sampling Point Value of 0x01-0x7F: 1 to 127. The hardware interprets this value as one more than the programmed value. Note: When the CAN clock (can_cclk) is 8 MHz, the reset value 0x0600 0A03 configures the CAN bit rate to 500 kbit/s.
7	Reserved		
15:8	NTSEG1	R/W	Nominal Time Period before Sampling Point Value of 0x01-0xFF: 1 to 255. The hardware interprets this value as one more than the programmed value.
24:16	NBRP	R/W	Nominal Bit Rate Prescaler 0x00-0x1FF is the divider value used to generate the bit time quantum oscillator frequency. The bit time is composed of multiples of this quantum. The effective value of bit ranges from 0 to 511. The hardware interprets this value as one more than the programmed value.
31:25	NSJW	R/W	Nominal (Re)Synchronization Jump Width Range of 0x00-0x7F: 0 to 127. The hardware interprets this value as one more than the programmed value.

20.7.8 Timestamp Counter Configuration (TSCC)

Offset address: 0x20

Reset value: 0x0000 0000

The internal 16-bit timestamp counter is configured. It handles internal and external timestamps.

Field	Name	R/W	Description
1:0	TSS	R/W	Timestamp Select 00: The timestamp counter value is always 0x0000. 01: The timestamp counter value increments according to TCP. 10: Use external timestamp counter value. 11: Same as “00”.
15:2	Reserved		
19:16	TCP	R/W	Timestamp Counter Prescaler 0x0-0xF: Configures the timestamp and timeout counter in units which are multiples of the CAN bit time [1…16]. The hardware interprets this value as one more than the programmed value. Note: Using CAN FD requires an external counter to generate the timestamp (TSS=10).
31:20	Reserved		

20.7.9 Timestamp Counter (TSCV)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	TSC	RC_W	Time Stamp Counter At the start of a frame (including both reception and transmission), the value of the internal/external timestamp counter is captured. When TSCC[TSS]=01, the timestamp counter increments in multiples of the CAN bit time [1…16] as configured by TSCC[TCP]. A counter overflow sets the interrupt flag IR[TSW]. A write access to it resets the counter to zero. When TSCC[TSS]=10, TSC reflects the value of the external timestamp counter, and a write access has no effect. Note: “Overflow” refers to the timestamp counter’s value changing from a non-zero value to zero, where this change is not caused by a write access to TSCV. A write operation to this register resets the timestamp counter.
31:16	Reserved		

20.7.10 Timeout Counter Configuration (TOCC)

Offset address: 0x28

Reset value: 0xFFFF 0000

Field	Name	R/W	Description
0	ETOC	R/W	Enable Timeout Counter 0: Disable timeout counter 1: Enable timeout counter

Field	Name	R/W	Description
			Note: For the information on the timeout function in CAN FD, refer to the “Timeout Counter” section.
2:1	TOS	R/W	Timeout Select When operating in continuous mode, a write to TOCV presets the counter to the value configured by TOCC[TOP] and the countdown continues. When the timeout counter is controlled by a FIFO, an empty FIFO presets the counter to the value configured by TOCC[TOP]. Decrementing begins when the first FIFO element is stored. 00: Continuous operation 01: Timeout controlled by Tx Event FIFO 10: Timeout controlled by Rx FIFO 0 11: Timeout controlled by Rx FIFO 1
15:3	Reserved		
31:16	TOP	R/W	Timeout Period The start value for the timeout counter (a decrementing counter), used to configure the timeout period.

20.7.11 Timeout Counter Value (TOCV)

Offset address: 0x2C

Reset value: 0x0000 FFFF

Field	Name	R/W	Description
15:0	TOC	RC_W	Timeout Counter The timeout counter decrements in multiples of the CAN bit time [1…16] as configured by TSCC[TCP]. When it decrements to zero, it sets the interrupt flag IR[TOO], and the timeout counter stops. The start and reset/restart conditions are configured via TOCC[TOS].
31:16	Reserved		

20.7.12 Error Counter Register (ECR)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	TEC	R	Transmit Error Counter Indicates the current state of the transmit error counter. The value ranges from 0 to 255. Note: When CCCR[ASM] is set, the CAN protocol controller does not increment the TEC and REC upon detection of a CAN protocol error, but the CEL is still incremented.
14:8	REC	R	Receive Error Counter Indicates the current state of the receive error counter. The value ranges from 0 to 127.
15	RP	R	Receive Error Passive 0: The Receive Error Counter is below the error passive level of 128. 1: The Receive Error Counter has reached the error passive level of 128.

Field	Name	R/W	Description
23:16	CEL	RC_W	CAN Error Logging This counter is incremented whenever a CAN protocol error causes the 8-bit Transmit Error Counter (TEC) or the 7-bit Receive Error Counter (REC) to be incremented. It is also incremented when the bus-off limit is reached. The counter is not incremented when only the RP bit is set without a change in the REC. The increment of the CEL occurs after the REC or TEC is incremented. A read access to the CEL resets this counter. The counter stops upon reaching 0xFF; the next increment of the TEC or REC sets the interrupt flag IR[ELO]. A read access to this register resets the CEL to zero.
31:24	Reserved		

20.7.13 Protocol Status Register (PSR)

Offset address: 0x44

Reset value: 0x0000 0707

Field	Name	R/W	Description
2:0	LEC	R/S	Last Error Code LED indicates the type of the last error on the CAN bus. This field is cleared to “0” when a message is transferred (received or transmitted) without error. 0 (No error): No error has occurred since the LEC was last reset by a successful message reception or transmission. 1 (Stuff Error): More than 5 consecutive bits of the same level were detected in a part of the received message, which is not allowed. 2 (Format Error): A fixed-format part of the received frame had the wrong format. 3 (AckError): CAN-transmitted message was not acknowledged by another node. 4 (Bit 1 Error): During message transmission (except in the arbitration field), the device attempted to send a recessive level (logic “1”), but the monitored bus value was dominant; 5 (Bit 0 Error): During message transmission (or in the Ack slot, active error flag, or overload flag), the device attempted to send a dominant level (logic “0” for data or identifier), but the monitored bus value was recessive. During bus-off recovery, this status is set each time a sequence of 11 consecutive recessive bits is monitored. This allows the CPU to monitor the progress of the bus-off recovery sequence (indicating the bus is not stuck dominant or persistently disturbed). 6 (CRC Error): The CRC checksum of the received message was incorrect. The CRC of the incoming message did not match the CRC calculated from the received data. 7 (No Change): Any read access to this register re-initializes the LEC to “7”. When the LEC shows a value of “7”, it means no new CAN bus event has been detected since the last read access to the Protocol Status Register by the CPU. Note: When a CAN FD format frame with the BRS flag set enters the data phase, the next CAN event (error or valid frame) is displayed in the DLEC other than in the LEC. Errors in the fixed

Field	Name	R/W	Description
			bit-stuff bits of the CAN FD CRC sequence are indicated as format errors other than stuff errors. Note: The bus-off recovery sequence (see ISO 11898-1:2015) cannot be shortened by setting or resetting CCCR[INIT]. If a device enters the bus-off state, it sets CCCR[INIT] autonomously, stopping all bus activity. Once the CPU clears CCCR[INIT], the device waits for 129 occurrences of bus idle (129 sequences of 11 consecutive recessive bits) before returning to normal operation. The error management counters are reset at the end of the bus-off recovery sequence. During the waiting period after CCCR[INIT] is reset, the Bit 0 Error code is written to PSR[LEC] each time a sequence of 11 consecutive recessive bits is monitored. This allows the CPU to easily check if the CAN bus is stuck dominant or persistently disturbed, and to monitor the bus-off recovery sequence. The ECR[REC] is used to count these sequences. A read access to this register sets the LEC to “111”.
4:3	ACT	R	Activity Status Monitors the CAN communication status of the module. 00: Synchronizing on a CAN node 01: Idle node, neither transmitting nor receiving 10: Receiving node, operating as a receiver 11: Transmitting node, operating as a transmitter Note: A protocol exception event sets ACT to 00.
5	EP	R	Error Passive Flag 0: CAN is in Error_Active state. Normally participates in bus communication and transmits an Active Error Flag upon detecting an error; 1: CAN is in Error_Passive state.
6	EW	R	Warning Status 0: Both error counters are below the Error_Warning limit of 96; 1: At least one error counter has reached or exceeded the Error_Warning limit of 96.
7	BO	R	Bus_Off Status 0: CAN is not in Bus_Off state 1: CAN has entered Bus_Off state
10:8	DLEC	R/S	Data Phase Last Error Code Indicates the type of the last error that occurred in the data phase of a CAN FD format frame with the BRS flag set. The encoding is identical to the LEC. This field is cleared when a CAN FD format frame with the BRS flag set is transferred (received or transmitted) without error. A read access to this register sets the DLEC to “111”.
11	RESI	RC_R	ESI flag of last received CAN FD Message This bit is set together with RFDF, independent of acceptance filtering. 0: No ESI flag set for the last received CAN FD message 1: ESI flag set for the last received CAN FD message. A read access to this register resets RESI.

Field	Name	R/W	Description
12	RBRS	RC_R	BRS flag of last received CAN FD Message This bit is set together with RFDF, independent of acceptance filtering. 0: No BRS flag set for the last received CAN FD message 1: BRS flag set for the last received CAN FD message A read access to this register resets RBRS
13	RFDF	RC_R	Received CAN FD Message This bit is set independent of acceptance filtering. 0: No CAN FD message received since the CPU reset this bit 1: A CAN FD format message with the FDF flag set received A read access to this register resets RFDF
14	PXE	RC_R	Protocol Exception Event 0: No protocol exception event occurring since the last read access 1: A protocol exception event occurred A read access to this register resets PXE
15	Reserved		
22:16	TDCV	R	Transceiver Delay Compensation Value Range: 0x00 – 0x7F. Defines the position of the secondary sampling point (SSP). It is defined by the sum of the measured delay from can_tx to can_rx and TDCR[TDCO]. During the data phase, the SSP position is the number of minimal time quanta (tq) from the start of the transmit bit to the secondary sampling point. Valid values are 0 to 127 mtq.
31:23	Reserved		

20.7.14 Transmitter Delay Compensation Register (TDCR)

Offset address: 0x48

Reset value: 0x0000 0000

Field	Name	R/W	Description
6:0	TDCF	R	Transmitter Delay Compensation Filter Window Length Range: 0x00-0x7F; Defines the minimum value for the secondary sampling point position. Dominant edges on can_rx that would cause the SSP position to occur earlier are ignored for transmitter delay measurement. This feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq.
7	Reserved		
14:8	TDCO	R	Transmitter Delay Compensation SSP Offset Range: 0x00-0x7F. This offset value defines the distance between the measured delay from can_tx to can_rx and the secondary sampling point. Valid values are 0 to 127 mtq.
31:15	Reserved		

20.7.15 Interrupt Register (IR)

Offset address: 0x50

Reset value: 0x0000 0000

When one of the conditions listed below is detected (a valid edge), the corresponding flag bit is set. These flag bits remain set until they are cleared by the host. A flag bit is cleared by writing “1” to its bit position. Writing “0”’ cannot clear flag bits. A hardware reset clears this register. The configuration of IE (Interrupt Enable) controls whether an interrupt is generated. The configuration of ILS (Interrupt Line Select) controls on which interrupt line the signal is asserted.

Field	Name	R/W	Description
0	RF0N	R/W	Rx FIFO 0 New Message 0: No new message written to Rx FIFO 0 1: A new message written to Rx FIFO 0
1	RF0W	R/W	Rx FIFO 0 Watermark Reached 0: The fill level of Rx FIFO 0 is below the watermark 1: The fill level of Rx FIFO 0 has reached or exceeded the watermark
2	RF0F	R/W	Rx FIFO 0 Full 0: Rx FIFO 0 not full 1: Rx FIFO 0 full
3	RF0L	R/W	Rx FIFO 0 Message Lost 0: No Rx FIFO 0 message lost 1: An Rx FIFO 0 message lost. This bit is also set after attempting to write to Rx FIFO 0 when its size is zero.
4	RF1N	R/W	Rx FIFO 1 New Message 0: No new message written to Rx FIFO 1 1: A new message written to Rx FIFO 1.
5	RF1W	R/W	Rx FIFO 1 Watermark Reached 0: The fill level of Rx FIFO 1 is below the watermark 1: The fill level of Rx FIFO 1 has reached or exceeded the watermark.
6	RF1F	R/W	Rx FIFO 1 Full 0: Rx FIFO 1 not full 1: Rx FIFO 1 full
7	RF1L	R/W	Rx FIFO 1 Message Lost 0: No Rx FIFO 1 message lost 1: An Rx FIFO 1 message lost. This bit is also set after attempting to write to Rx FIFO 1 when its size is zero.
8	HPM	R/W	High Priority Message 0: No high-priority message received 1: A high-priority message received
9	TC	R/W	Transmission Completed 0: No transmission completed 1: Transmission completed
10	TCF	R/W	Transmission Cancellation Finished 0: No transmission cancellation completed 1: Transmission cancellation completed
11	TFE	R/W	Tx FIFO Empty 0: The Tx FIFO is not empty 1: The Tx FIFO is empty

Field	Name	R/W	Description
12	TEFN	R/W	Tx Event FIFO New Entry 0: No change in the Tx Event FIFO 1: The Tx Handler has written a Tx Event FIFO element
13	TEFW	R/W	Tx Event FIFO Watermark Reached 0: The fill level of the Tx Event FIFO is below the watermark 1: The fill level of the Tx Event FIFO has reached or exceeded the watermark
14	TEFF	R/W	Tx Event FIFO Full 0: The Tx Event FIFO is not full 1: The Tx Event FIFO is full
15	TEFL	R/W	Tx Event FIFO Element Lost 0: No Tx Event FIFO element lost 1: A Tx Event FIFO element lost. This bit is also set after attempting to write to the Tx Event FIFO when its size is zero.
16	TSW	R/W	Timestamp Wraparound 0: The Timestamp Counter has not wrapped around 1: The Timestamp Counter has wrapped around
17	MRAF	R/W	Message RAM Access Failure This flag is set when the Receive Handler: Fails to complete the acceptance filtering or storage of a received message before the arbitration field of the next message is received. In this case, acceptance filtering or message storage is aborted, and the Rx Handler starts processing the next message. Is unable to write a message to the Message RAM. In this case, message storage is aborted. In both cases above, the FIFO write index is not updated, and the new data flag of a dedicated receive buffer is not set. The partially stored message will be overwritten when the next message is stored to this location. This flag bit is also set when the Tx Handler fails to read a message from the Message RAM in time. In this case, the message transmission is aborted. If a Tx Handler access failure occurs, the CAN module switches to Restricted Operation Mode. To exit Restricted Operation Mode, the host CPU must reset CCCR[ASM]. 0: No Message RAM access failure 1: A Message RAM access failure
18	TOO	R/W	Timeout Occurred 0: No timeout 1: The timeout period has been reached
19	DRX	R/W	Message Stored to Dedicated Rx Buffer This flag is set whenever a received message is stored to a Dedicated Receive Buffer. 0: No receive buffer updated 1: At least one received message has been stored to a receive buffer.
21:20	Reserved		
22	ELO	R/W	Error Logging Overflow 0: The CAN error logging counter has not overflowed 1: The CAN error logging counter has overflowed
23	EP	R/W	Error Passive Status 0: No change in the error-passive status 1: Change in the error-passive status

Field	Name	R/W	Description
24	EW	R/W	Warning Status 0: No change in the error-warning status 1: Change in error-warning status
25	BO	R/W	Bus_Off Status 0: No change in Bus-Off status 1: Change in the Bus-Off status
26	WDI	R/W	Watchdog Interrupt 0: No Message RAM Watchdog event 1: A Message RAM Watchdog event has occurred due to a missing READY signal
27	PEA	R/W	Protocol Error in Arbitration Phase (Nominal bit time is used) 0: No protocol error has occurred in the Arbitration Phase 1: A protocol error was detected in the Arbitration Phase (PSR[LEC]≠0, 7)
28	PED	R/W	Protocol Error in Data Phase (Data Bit Time is used) 0: No protocol error has occurred in the Data Phase 1: A protocol error was detected in the Data Phase (PSR[DLEC]≠0, 7)
29	ARA	R/W	Access to Reserved Address 0: No access to a reserved address 1: An access to a reserved address
31:30	Reserved		

20.7.16 Interrupt Enable (IE)

Offset address: 0x54

Reset value: 0x0000 0000

The settings in the Interrupt Enable Register determine which status changes in the Interrupt Register will signal an event via the interrupt lines.

0: Disable interrupt

1: Enable interrupt

Field	Name	R/W	Description
0	RF0NE	R/W	Rx FIFO 0 New Message Interrupt Enable
1	RF0WE	R/W	Rx FIFO 0 Watermark Reached Interrupt Enable
2	RF0FE	R/W	RX FIFO 0 Full Interrupt Enable
3	RF0LE	R/W	Rx FIFO 0 Message Lost Interrupt Enable
4	RF1NE	R/W	Rx FIFO 1 New Message Interrupt Enable
5	RF1WE	R/W	Rx FIFO 1 Watermark Reached Interrupt Enable
6	RF1FE	R/W	RX FIFO 1 Full Interrupt Enable
7	RF1LE	R/W	Rx FIFO 1 Message Lost Interrupt Enable
8	HPME	R/W	High Priority Message Interrupt Enable
9	TCE	R/W	Transmission Completed Interrupt Enable
10	TCFE	R/W	Transmission Cancellation Finished Interrupt Enable
11	TFEE	R/W	TX FIFO Empty Interrupt Enable

Field	Name	R/W	Description
12	TEFNE	R/W	Tx Event FIFO New Entry Interrupt Enable
13	TEFWE	R/W	Tx FIFO Watermark Reached Interrupt Enable
14	TEFFE	R/W	TX Event FIFO Full Interrupt Enable
15	TEFLE	R/W	Tx Event FIFO Element Lost Interrupt Enable
16	TSWE	R/W	Timestamp Wraparound Interrupt Enable
17	MRAFE	R/W	Message RAM Access Failure Interrupt Enable
18	TOOE	R/W	Timeout Occurred Interrupt Enable
19	DRXE	R/W	Message Stored to Dedicated Rx Buffer Interrupt Enable
21:20	Reserved		
22	ELOE	R/W	Error Logging Overflow Interrupt Enable
23	EPE	R/W	Error Passive Interrupt Enable
24	EWE	R/W	Warning Status Interrupt Enable
25	BOE	R/W	Bus_Off Status Interrupt Enable
26	WDIE	R/W	Watchdog Interrupt Enable
27	PEAE	R/W	Protocol Error in Arbitration Phase Enable
28	PEDE	R/W	Protocol Error in Data Phase Enable
29	ARAE	R/W	Access to Reserved Address Enable
31:30	Reserved		

20.7.17 Interrupt Line Select

Offset address: 0x58

Reset value: 0xXXXX XXXX

The Interrupt Line Select register assigns interrupts generated by specific interrupt flags in the interrupt register to one of the two module interrupt lines. To generate an interrupt, the corresponding interrupt line must be enabled via ILE[EINT0] and ILE[EINT1].

0: Interrupt applied to Interrupt Line CAN_INT0

1: Interrupt applied to Interrupt Line CAN_INT1

Field	Name	R/W	Description
0	RF0NL	R/W	Rx FIFO 0 New Message Interrupt Line Select
1	RF0WL	R/W	Rx FIFO 0 Watermark Reached Interrupt Line Select
2	RF0FL	R/W	RX FIFO 0 Full Interrupt Line Select
3	RF0LL	R/W	Rx FIFO 0 Message Lost Interrupt Line Select
4	RF1NL	R/W	Rx FIFO 1 New Message Interrupt Line Select
5	RF1WL	R/W	Rx FIFO 1 Watermark Reached Interrupt Line Select

Field	Name	R/W	Description
6	RF1FL	R/W	RX FIFO 1 Full Interrupt Line Select
7	RF1LL	R/W	Rx FIFO 1 Message Lost Interrupt Line Select
8	HPML	R/W	High Priority Message Interrupt Line Select
9	TCL	R/W	Transmission Completed Interrupt Line Select
10	TCFL	R/W	Transmission Cancellation Finished Interrupt Line Select
11	TFEL	R/W	TX FIFO Empty Interrupt Line Select
12	TEFNL	R/W	Tx Event FIFO New Entry Interrupt Line Select
13	TEFWL	R/W	Tx Event FIFO Watermark Reached Interrupt Line Select
14	TEFFL	R/W	Tx Event FIFO Full Interrupt Line Select
15	TEFLL	R/W	Tx Event FIFO Element Lost Interrupt Line Select
16	TSWL	R/W	Timestamp Wraparound Interrupt Line Select
17	MRAFL	R/W	Message RAM Access Failure Interrupt Line Select
18	TOOL	R/W	Timeout Occurred Interrupt Line Select
19	DRXL	R/W	Message Stored to Dedicated Rx Buffer Interrupt Line Select
21:20	Reserved		
22	ELOL	R/W	Error Logging Overflow Interrupt Line Select
23	EPL	R/W	Error Passive Status Interrupt Line Select
24	EWL	R/W	Warning Status Interrupt Line Select
25	BOL	R/W	Bus_Off Status Interrupt Line Select
26	WDIL	R/W	Watchdog Interrupt Line Select
27	PEAL	R/W	Protocol Error Interrupt Line Select in Arbitration Phase
28	PEDL	R/W	Protocol Error Interrupt Line Select in Data Phase
29	ARAL	R/W	Access to Reserved Address Interrupt Line Select
31:30	Reserved		

20.7.18 Interrupt Line Enable

Offset address: 0x5C

Reset value: 0x0000 0000

The two interrupt lines connected to the CPU can be individually enabled/disabled by programming EINT0 and EINT1 bits, respectively.

Field	Name	R/W	Description
0	EINT0	R/W	Enable Interrupt Line 0 0: Disable Interrupt Line CAN_INT0 1: Enable Interrupt Line CAN_INT0

Field	Name	R/W	Description
1	EINT1	R/W	Enable Interrupt Line 1 0: Disable Interrupt Line CAN_INT1 1: Enable Interrupt Line CAN_INT1
31:2	Reserved		

20.7.19 Global Filter Configuration (GFC)

Offset address: 0x80

Reset value: 0x0000 0000

This is the global setting for message ID filtering.

Field	Name	R/W	Description
0	RRFE	R	Reject Extended Remote Frames 0: Filters remote frames with 29-bit extended IDs 1: Rejects all remote frames with 29-bit extended IDs.
1	RRFS	R	Reject Standard Remote Frames 0: Filters remote frames with 11-bit standard IDs 1: Rejects all remote frames with 11-bit standard IDs.
3:2	ANFE	R	Accept Non-matching Frames Extended Defines how a received message with a 29-bit ID that does not match any element in the filter list is handled. 00: Accept in Rx FIFO 0 01: Accept in Rx FIFO 1 10: Reject 11: Reject
5:4	ANFS	R	Accept Non-matching Standard Frames Defines how a received message with a 11-bit ID that does not match any element in the filter list is handled. 00: Accept in Rx FIFO 0 01: Accept in Rx FIFO 1 10: Reject 11: Reject
31:6	Reserved		

20.7.20 Standard ID Filter Configuration (SIDFC)

Offset address: 0x84

Reset value: 0x0000 0000

This is the configuration for filtering 11-bit standard message IDs. The standard ID filter element configuration controls the filtering path for standard messages.

Field	Name	R/W	Description
1:0	Reserved		
15:2	FLSSA	R	Standard Filter List Start Address The start address (32-bit word address) of the standard message ID Filter List in the Message RAM.
23:16	LSS	R	Standard List Size 0: No Standard Message ID Filters 1 to 128: Number of Standard Message ID Filter Elements Values greater than 128 are interpreted as 128.

Field	Name	R/W	Description
31:24	Reserved		

20.7.21 Extended ID Filter Configuration (XIDFC)

Offset address: 0x88

Reset value: 0x0000 0000

This is the configuration for filtering 29-bit extended message IDs. The extended ID filter element configuration controls the filtering path for standard messages.

Field	Name	R/W	Description
1:0	Reserved		
15:2	FLESA	R	Extended Filter List Start Address The start address (32-bit word address) of the extended message ID Filter List in the Message RAM.
22:16	LSE	R	Extended List Size 0: No Extended Message ID Filters 1 to 64: Number of Extended Message ID Filter Elements Values greater than 64 are interpreted as 64.
31:23	Reserved		

20.7.22 Extended ID and Mask (XIDAM)

Offset address: 0x90

Reset value: 0x1FFF FFFF

Field	Name	R/W	Description
28:0	EIDM	R	Extended ID Mask Acceptance filtering for extended frames: The AND 312peration will be implemented between the extended ID and the Message ID of the received frames. This is used to mask the 29-bit ID in protocols like SAE J1939. When the reset values of all bits are “1”, the mask is invalid.
31:29	Reserved		

20.7.23 High Priority Message Status (HPMS)

Offset address: 0x94

Reset value: 0x0000 0000

This register is updated whenever a Message ID Filter element, configured to generate a priority event, finds a match. This can be used to monitor the status of incoming high-priority messages and enables fast access to them.

Field	Name	R/W	Description
5:0	BIDX	R	Buffer Index The index of the Rx FIFO element where the message is stored. This is only valid if MSI[1] = 1.
7:6	MSI	R	Message Storage Indicator 00: No FIFO selected 01: FIFO message lost 10: Message stored in FIFO 0 11: Message stored in FIFO 1

Field	Name	R/W	Description
14:8	FIDX	R	Filter Index The index of the matching filter element. The range is from 0 to SIDFC[LSS]-1 or XIDFC[LSE]-1.
15	FLST	R	Filter List Indicates the list of filter matching filter elements. 0: List of Standard Filters 1: List of Extended Filters.
31:16	Reserved		

20.7.24 New Data 1 (NDAT1)

Offset address: 0x98

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	Ndy	R/W	New Data (y=0…31) This register holds the new data flags for Rx Buffers 0 to 31. The flags are set when the corresponding Rx Buffer is updated by a received frame. These flag bits remain set until they are cleared by the host. A flag bit is cleared by writing “1” to its bit position. Writing “0”’ cannot clear flag bits. A hardware reset clears this register. 0: Rx Buffer not updated 1: Rx Buffer updated with a new message

20.7.25 New Data 2 (NDAT2)

Offset address: 0x9C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	ND(y+32)	R/W	New Data (y=0…31) This register holds the new data flags for Rx Buffers 32 to 63. The flags are set when the corresponding Rx Buffer is updated by a received frame. These flag bits remain set until they are cleared by the host. A flag bit is cleared by writing “1” to its bit position. Writing “0”’ cannot clear flag bits. A hardware reset clears this register. 0: Rx Buffer not updated 1: Rx Buffer updated with a new message

20.7.26 Rx FIFO 0 Configuration (RXF0C)

Offset address: 0xA0

Reset value: 0x0000 0000

Field	Name	R/W	Description
1:0	Reserved		
15:2	F0SA	R	Rx FIFO 0 Start Address The start address of Rx FIFO 0 in Message RAM (32-bit word address)
22:16	F0S	R	Rx FIFO 0 Size 0: No Rx FIFO 0 1 to 64: Number of Rx FIFO 0 elements Values greater than 64 are interpreted as 64 Index of Rx FIFO 0 elements ranges from 0 to F0S -1
23	Reserved		

Field	Name	R/W	Description
30:24	F0WM	R	Rx FIFO 0 Watermark 0: Disable watermark interrupt 1~64: Level of Rx FIFO 0 watermark interrupt (IR[RF0W]) More than 64: Disable watermark interrupt
31	F0OM	R	FIFO 0 Operation Mode FIFO 0 can operate in Block Mode or Overwrite Mode. 0: FIFO 0 Block Mode 1: FIFO 0 Overwrite Mode

20.7.27 Rx FIFO 0 Status (RXF0S)

Offset address: 0xA4

Reset value: 0x0000 0000

Field	Name	R/W	Description
6:0	F0FL	R	Rx FIFO 0 Fill Level Number of elements stored in Rx FIFO 0, ranging from 0 to 64.
7	Reserved		
13:8	F0GI	R	Rx FIFO 0 Read Index Rx FIFO 0 read index pointer, ranging from 0 to 63.
15:14	Reserved		
21:16	F0PI	R	Rx FIFO 0 Write Index Rx FIFO 0 write index pointer, ranging from 0 to 63.
23:22	Reserved		
24	F0F	R	Rx FIFO 0 Full 0: Rx FIFO 0 not full 1: Rx FIFO 0 full
25	RF0L	R	Rx FIFO 0 Message Lost This bit is a copy of interrupt flag IR[RF0L]. It is reset when IR[RF0L] is reset. 0: No Rx FIFO 0 message lost 1: Rx FIFO 0 message lost. This bit is also set after attempting to write to Rx FIFO 0 when its fill level is zero. Note: When RXF0C[F0OM] = 1, overwriting the oldest message does not set this flag.
31:26	Reserved		

20.7.28 Rx FIFO 0 Acknowledge (RXF0A)

Offset address: 0xA8

Reset value: 0x0000 0000

Field	Name	R/W	Description
5:0	F0AI	R/W	Rx FIFO 0 Acknowledge Index After the host has read one message or a sequence of messages from Rx FIFO 0, it must write the Buffer Index of the last element read from Rx FIFO 0 to F0AI. This sets the Rx FIFO 0 read index RXF0S[F0GI] to F0AI + 1 and updates the FIFO 0 fill level RXF0S[F0FL].
31:6	Reserved		

20.7.29 Rx Buffer Configuration (RXBC)

Offset address: 0xAC

Reset value: 0x0000 0000

Field	Name	R/W	Description
1:0	Reserved		
15:2	RBSA	R	Rx Buffer Start Address Configures the start address (32-bit word address) of the Rx Buffer section in the Message RAM.
31:16	Reserved		

20.7.30 Rx FIFO 1 Configuration (RXF1C)

Offset address: 0xB0

Reset value: 0x0000 0000

Field	Name	R/W	Description
1:0	Reserved		
15:2	F1SA	R	Rx FIFO 1 Start Address The start address of Rx FIFO 1 in Message RAM (32-bit word address)
22:16	F1S	R	Rx FIFO 1 Size 0: No Rx FIFO 1 1 to 64: Number of Rx FIFO 1 elements Values greater than 64 are interpreted as 64 Index of Rx FIFO 1 elements ranges from 0 to F1S -1
23	Reserved		
30:24	F1WM	R	Rx FIFO 1 Watermark 0: Disable watermark interrupt 1~64: Level of Rx FIFO 1 watermark interrupt (IR[RF1W]) More than 64: Disable watermark interrupt
31	F1OM	R	FIFO 1 Operation Mode FIFO 1 can operate in Block Mode or Overwrite Mode 0: FIFO 1 in Block Mode 1: FIFO 1 in Overwrite Mode

20.7.31 Rx FIFO 1 Status (RXF1S)

Offset address: 0xB4

Reset value: 0x0000 0000

Field	Name	R/W	Description
6:0	F1FL	R	Rx FIFO 1 Fill Level Number of elements stored in Rx FIFO 1, ranging from 0 to 64.
7	Reserved		
13:8	F1GI	R	Rx FIFO 1 Read Index Rx FIFO 1 read index pointer, ranging from 0 to 63.
15:14	Reserved		
21:16	F1PI	R	Rx FIFO 1 Write Index Rx FIFO 1 write index pointer, ranging from 0 to 63.
23:22	Reserved		

Field	Name	R/W	Description
24	F1F	R	Rx FIFO 1 Full 0: Rx FIFO 1 not full 1: Rx FIFO 1 full Bit 21: 16 F1PI [5:0]: Rx FIFO 1 write index Rx FIFO 1 write index pointer, ranging from 0 to 63.
25	RF1L	R	Rx FIFO 1 Message Lost This bit is a copy of interrupt flag IR[RF1L]. It is reset when IR[RF1L] is reset. 0: No Rx FIFO 1 message lost 1: Rx FIFO 1 message lost. This bit is also set after attempting to write to Rx FIFO 1 when its fill level is zero. Note: When RXF1C[F1OM] = 1, overwriting the oldest message does not set this flag.
31:26	Reserved		

20.7.32 Rx FIFO 1 Acknowledge (RXF1A)

Offset address: 0xB8

Reset value: 0x0000 0000

Field	Name	R/W	Description
5:0	F1AI	R/W	Rx FIFO 1 Acknowledge Index After the host has read one message or a sequence of messages from Rx FIFO 1, it must write the Buffer Index of the last element read from Rx FIFO 1 to F1AI. This sets the Rx FIFO 1 read index RXF1S[F1GI] to F1AI + 1 and updates the FIFO 1 fill level RXF1S[F1FL].
31:6	Reserved		

20.7.33 Rx Buffer/ FIFO Element Size Configuration (RXESC)

Offset address: 0xBC

Reset value: 0x0000 0000

Configures the number of data bytes for elements belonging to an Rx Buffer / Rx FIFO. A data field size greater than 8 bytes is only used for CAN FD operation.

Field	Name	R/W	Description
2:0	F0DS	R	Rx FIFO 0 Data Field Size 000: 8-byte data field 001: 12-byte data field 0 10: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field Note: If a received CAN frame has a data field size larger than the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes configured by RXESC is stored into the Rx Buffer or Rx FIFO element. The remaining data field of the frame is ignored.

Field	Name	R/W	Description
3	Reserved		
6:4	F1DS	R	Rx FIFO 1 Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field
7	Reserved		
10:8	RBDS	R	Rx Buffer Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field
31:11	Reserved		

20.7.34 Tx Buffer Configuration (TXBC)

Offset address: 0xC0

Reset value: 0x0000 0000

Field	Name	R/W	Description
1:0	Reserved		
15:2	TBSA	R	Tx Buffers Start Address Start address of the Tx Buffer section in the Message RAM. Note: The sum of TFQS and NDTB may not be greater than 32. Erroneous configurations are not checked. The Tx Buffer section in the Message RAM starts with the dedicated Tx Buffers.
21:16	NDTB	R	Number of dedicated Tx buffers 0: No dedicated Tx buffers 1 to 32: Number of dedicated Tx buffers Values greater than 32 are interpreted as 32
23:22	Reserved		
29:24	TFQS	R	Transmit FIFO/Queue Size 0: No Tx FIFO/Queue 1 to 32: Number of Tx buffers for FIFO/Queue Values greater than 32 are interpreted as 32
30	TFQM	R	Tx FIFO/Queue Mode 0 : Tx FIFO mode 1 : Tx Queue Mode
31	Reserved		

20.7.35 Tx FIFO / Queue Status (TXFQS)

Offset address: 0xC4

Reset value: 0x0000 0000

The Tx FIFO/Queue status is related to the pending Tx requests listed in register TXBRP. Therefore, the effect of adding/canceling requests may be delayed due to an ongoing Tx scan (TXBRP has not been updated yet).

Field	Name	R/W	Description
5:0	TFFL	R	Tx FIFO Free Level The number of consecutive free Tx FIFO elements starting from TFGI, ranging from 0 to 32. Reads as 0 when configured to Tx queue operation mode (TXBC[TFQM]=1). Note: In configurations mixing dedicated Tx Buffers with a Tx FIFO or Tx Queue, the write and read indices indicate the Tx Buffer number starting from the first dedicated Tx Buffer. Example: For a configuration with 12 dedicated Tx Buffers and a Tx FIFO of 20 buffers, write index 15 points to the fourth buffer of the Tx FIFO.
7:6	Reserved		
12:8	TFGI	R	Tx FIFO Read Index Tx FIFO read index pointer, ranging from 0 to 31. Reads as 0 when configured to Tx queue operation mode (TXBC[TFQM]=1).
15:13	Reserved		
20:16	TFQPI	R	Tx FIFO/Queue Write Index Tx FIFO/Queue write index pointer, ranging from 0 to 31.
21	TFQF	R	Tx FIFO/Queue Full 0: Tx FIFO / Queue not full 1: Tx FIFO / Queue full
31:22	Reserved		

20.7.36 Tx Buffer Element Size Configuration (TXESC)

Offset address: 0xC8

Reset value: 0x0000 0000

Configures the number of data bytes in Tx buffer element. A data field size greater than 8 bytes is only used for CAN FD operation.

Field	Name	R/W	Description
2:0	TBDS	R	Tx Buffer Data Field Size 000: 8-byte data field 001: 12-byte data field 010: 16-byte data field 011: 20-byte data field 100: 24-byte data field 101: 32-byte data field 110: 48-byte data field 111: 64-byte data field Note: If the data length code (DLC) of a Tx Buffer element is configured with a value higher than the Tx Buffer data field size (TXESC[TBDS]), the undefined bytes of the Tx Buffer are transmitted as 0xCC (pad bytes).
31:3	Reserved		

20.7.37 Tx Buffer Request Pending (TXBRP)

Offset address: 0xCC

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	TRPy	R	Pending Request for Tx Buffer y (y=0…31) Each Tx Buffer has its own Transmission Request Pending bit. These bits are set via the TXBAR register. They are reset after the requested transmission is completed or canceled via the TXBCR register. TXBRP bits are only set for Tx Buffers configured via TXBC. After a TXBRP bit is set, a Tx scan is initiated to check for the pending Tx request with the highest priority (the Tx Buffer with the lowest Message ID). Canceling a request resets the corresponding Transmission Request Pending bit in the TXBRP register. If the transmission has already started at the time of the cancellation request, the reset occurs at the end of the transmission, regardless of whether it was successful or not. The Cancellation Request bit is reset immediately after the corresponding TXBRP bit is reset. The completion of the cancellation is signaled via TXBCF after the cancellation request: Upon successful transmission, along with the corresponding TXBTO bit When the transmission has not yet started at the cancellation point When the transmission is aborted due to arbitration loss When an error occurs during frame transmission In DAR mode, all transmissions are automatically canceled if they are unsuccessful. For all unsuccessful transmissions, the corresponding TXBCF bit is set. 0: No transmission request pending 1: Transmission request pending Note: TXBRP bits set during an ongoing Tx scan are not considered for that specific Tx scan. If a cancellation is requested for such a Tx Buffer, this add request is canceled immediately, and the corresponding TXBRP bit is reset.

20.7.38 Tx Buffer Add Request (TXBAR)

Offset address: 0xD0

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	ARy	R/W	Add Request for Tx Buffer y (y=0…31) Each Tx Buffer has its own Add Request bit. Writing 1 sets the corresponding Add Request bit; writing 0 has no effect. This allows the host to set transmission requests for multiple Tx Buffers with a single write to TXBAR. TXBAR bits are only set for Tx Buffers configured via TXBC. These bits are reset immediately when no Tx scan is running; otherwise, they remain set until the Tx scan procedure is completed. 0: No transmission request added 1: Transmission request added Note: If an Add Request is applied to a Tx Buffer that already has a pending transmission request (its corresponding TXBRP bit is already set), this Add Request is ignored.

20.7.39 Tx Buffer Cancellation Request (TXBCR)

Offset address: 0xD4

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	CRy	R/W	Cancellation Request for Tx Buffer y (y=0…31) Each Tx Buffer has its own Cancellation Request bit. Writing 1 sets the corresponding Cancellation Request bit; writing 0 has no effect. This allows the host to set cancellation requests for multiple Tx Buffers with a single write to TXBCR. TXBCR bits are only set for Tx Buffers configured via TXBC. These bits remain set until the corresponding bits in TXBRP are reset. 0: No cancellation pending 1: Cancellation pending

20.7.40 Transmission Occurred for Tx Buffer (TXBTO)

Offset address: 0xD8

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	TOy	R	Transmission Occurred for Tx Buffer y (y=0…31) Each Tx Buffer has its corresponding “Transmission Occurred” bit. These bits are set when the corresponding TXBRP bit is cleared after a successful transmission. They are reset when a new transmission is requested by writing a ‘1’ to the corresponding bit in the TXBAR register. 0: No transmission 1: Transmission occurred

20.7.41 Tx Buffer Cancellation Finished (TXBCF)

Offset address: 0xDC

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	CFy	R	Cancellation Finished for Tx Buffer y (y=0…31) Each Tx Buffer has its own Cancellation Finished bit. These bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. If the corresponding TXBRP bit was not set at the time of cancellation, the CF bit is set immediately. They are reset when a new transmission is requested by writing a ‘1’ to the corresponding bit in the TXBAR register. 0: Tx Buffer cancellation not finished 1: Tx Buffer cancellation finished

20.7.42 Tx Buffer Transmission Interrupt Enable (TXBTIE)

Offset address: 0xE0

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	TIEy	R/W	Transmission Interrupt Enable for Tx Buffer y (y=0…31) Each Tx Buffer has its corresponding Transmission Interrupt Enable bit. 0: Disable 1: Enable

20.7.43 Tx Buffer Cancellation Finished Interrupt Enable (TXBCIE)

Offset address: 0xE4

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	CFIEy	R/W	Cancellation Finished Interrupt Enable for Tx Buffer y (y=0…31) Each Tx Buffer has its corresponding Cancellation Finished Interrupt Enable bit. 0: Disable 1: Enable

20.7.44 Tx Event FIFO Configuration (TXEFC)

Offset address: 0xF0

Reset value: 0x0000 0000

Field	Name	R/W	Description
1:0	Reserved		
15:2	EFSA	R	Event FIFO Start Address The start address of Tx Event FIFO in Message RAM (32-bit word address)
21:16	EFS	R	Event FIFO Size 0: Disable Tx event FIFO 1 to 32: Number of Tx event FIFO elements Values greater than 32 are interpreted as 32 Index of Tx event FIFO elements ranges from 0 to EFS -1
23:22	Reserved		
29:24	EFWM	R	Event FIFO Watermark 0: Disable watermark interrupt 1~32: Level of Tx Even FIFO Watermark Interrupt (IR[TEFW]) More than 32: Disable watermark interrupt
31:30	Reserved		

20.7.45 Tx Event FIFO Status (TXEFS)

Offset address: 0xF4

Reset value: 0x0000 0000

Field	Name	R/W	Description
5:0	EFFL	R	Event FIFO Fill Level Number of elements stored in Tx Event FIFO, ranging from 0 to 32.
7:6	Reserved		
12:8	EFGI	R	Event FIFO Read Index Tx Event FIFO read index pointer, ranging from 0 to 31.
15:13	Reserved		
20:16	EFPI	R	Event FIFO Write Index Tx Event FIFO write index pointer, ranging from 0 to 31.
23:21	Reserved		
24	EFF	R	Event FIFO Full 0: The Tx Event FIFO is not full 1: The Tx Event FIFO is full
25	TEFL	R	Tx Event FIFO Element Lost This bit is a copy of interrupt flag IR[TEFL]. It is reset when IR[TEFL] is reset. 0: No Tx Event FIFO element lost 1: A Tx Event FIFO element lost. This bit is also set after attempting to write to the Tx Event FIFO when its size is zero.

Field	Name	R/W	Description
31:26	Reserved		

20.7.46 Tx Event FIFO Acknowledge (TXEFA)

Offset address: 0xF8

Reset value: 0x0000 0000

Field	Name	R/W	Description
4:0	EFAI	R/W	Event FIFO Acknowledge Index After the host has read one element or a sequence of elements from the Tx Event FIFO, it must write the index of the last element read from the Tx Event FIFO to EFAI. This sets the Tx Event FIFO read index TXEFS[EFGI] to EFAI + 1 and updates the Event FIFO fill level TXEFS[EFFL].
31:5	Reserved		

Note: The corresponding registers of CAN_ERM module are shown as follows.

20.7.47 ERM and Low-Power Control (ERM_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ECC_EN	R/W	Enable SRAM ECC Check and Global Interrupt for SRAM ECC Check 0: Disable CAN SRAM ECC Check and Global ECC Interrupt 1: Enable CAN SRAM ECC Check and Global ECC Interrupt
1	BEC_INT_EN	R/W	Bit Error Correction Interrupt Enable 0: Disable 1: Enable Note: To ensure correct generation of interrupts, ensure ECC_EN = 1.
2	BEU_INT_EN	R/W	Bit Error Non-Correction Interrupt Enable 0: Disable 1: Enable Note: To ensure correct generation of interrupts, ensure ECC_EN = 1.
3	WAKE_INT_EN	R/W	CAN Wake-up Interrupt Enable 0: Disable 1: Enable
4	WAKE_EVENT_EN	R/W	CAN Wake-up Event Enable 0: Disable 1: Enable
31:5	Reserved		

20.7.48 ERM and Low-Power Status (ERM_STS)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	BEC_FLAG	R/W	Single-Bit Error Detected and Correctable 0: No error detected 1: Single-bit error detected
1	BEU_FLAG	R/W	Multi-bit Error Detected and Uncorrectable 0: No error detected or only single-bit error detected 1: Multi-bit error detected
2	CAN_WAKEUP_FLAG	R/W	CAN Wake-Up Flag 0: No CAN wake-up detected 1: CAN wake-up detected
31:3	Reserved		

20.7.49 ERM Error Address Register (ERM_ECCLOG)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
8:0	ERR_ADDR	R/W	Error Location Address Note: This address is the corresponding CAN SRAM offset address shifted right by 2 bits.
31:9	Reserved		

21 Analog-to-Digital Converter (ADC)

21.1 Introduction

This series product has one ADC with 12-bit precision and 19 channels, including 16 external channels and 3 internal channels. It supports single, continuous or intermittent A/D conversion modes for each channel. ADC conversion results can be left-aligned or right-aligned and stored in 16-bit data registers.

21.2 Main Characteristics

(1) ADC power supply requirements: From 2.75 V to 5.5 V

(2) ADC input range: $V_{SSA} \leqslant V_{IN} \leqslant V_{DDA}$

(3) 12-bit resolution

(4) Conversion mode
- Single conversion mode
- Continuous conversion mode
- Intermittent mode

(5) Analog input channel category
- External GPIO input channel
- One internal temperature sensor (V_{SENSE}) input channel
- One internal reference voltage (V_{REFINT}) input channel

(6) High performance
- 12-bit, 10-bit, 8-bit or 6-bit configurable resolution
- Self-calibration
- Programmable sampling time
- Data alignment
- DMA supported

(7) Low power
- Low-power operation reduces PLCK frequency and maintains optimum ADC performance
- Automatic delay mode: Run in PCLK low speed, to prevent ADC over-limit
- Automatic shutdown mode: ADC can power off automatically at other times except during conversion period

(8) Interrupt
- End of conversion interrupt
- End of sequence conversion interrupt
- End of sampling phase interrupt

- ADC ready interrupt
- Overrun interrupt
- Analog watchdog state reset interrupt

(9) Trigger mode

- Internal signal trigger generated by on-chip timer

21.3 Functional Description

21.3.1 ADC pin and internal signal

Table 79 ADC Internal Signal

Name	Description	Signal type
TMRx_TRG	Internal information from timer	Input
V_{SENSE}	Output voltage of internal temperature sensor	Input
V_{REFINT}	Output of internal reference voltage	Input

Table 80 ADC Pins

Name	Description	Signal type
V_{DDA}	Analog power supply, positive ADC reference voltage, $V_{DDA} \geqslant V_{DD}$	Input, analog power supply
V_{SSA}	Analog power ground, $V_{SSA}=V_{SS}$	Input, analog power ground
ADC_IN[15:0]	16-channel analog input	Analog input signal

21.3.2 Calibration

The function of calibration is to eliminate the offset error of A/D conversion of each chip, so calibration should be conducted before A/D conversion, and ADC module cannot be used during calibration.

Calibration configuration process:

- Configure ADCEN bit of register ADC_CTRL to 0, and disable ADC
- Configure CAL bit of register ADC_CTRL to 1, and enable calibration
- After calibration is completed, CAL bit is automatically cleared by hardware
- The calibration factor is read in CDATA[6:0] bit of register ADC_DATA

21.3.3 ADC conversion mode

21.3.3.1 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.

When CMODESEL bit of configuration register ADC_CFG1 is 0, ADC is set to single conversion mode; ADC conversion can be enabled by setting STARTCEN bit of configuration register ADC_CTRL to 1 by software or by hardware-triggered event.

After the conversion of each channel, the converted data will be stored in the 16-bit ADC_DATA register, EOCFLG bit will be set to 1, and if EOCIEN bit is set to 1, an interrupt will be generated. After the channel sequence conversion, EOSEQFLG bit will be set to 1, and if EOSEQIEN bit is set to 1, an interrupt will be generated.

21.3.3.2 Continuous conversion mode

In this mode, for single channel, continuous conversion is only conducted for this channel; for multiple channels, continuous conversion is only conducted for this group of channels.

When CMODESEL bit of configuration register ADC_CFG1 is set to 1, ADC is set to continuous conversion mode; ADC conversion can be enabled by setting STARTCEN bit of configuration register ADC_CTRL to 1 by software or by hardware-triggered event.

After the conversion of each channel, the converted data will be stored in the 16-bit ADC_DATA register, EOCFLG bit will be set to 1, and if EOCIEN bit is set to 1, an interrupt will be generated. After the channel sequence conversion, EOSEQFLG bit will be set to 1, and if EOSEQIEN bit is set to 1, an interrupt will be generated.

21.3.3.3 Intermittent mode

When DISCEN bit of configuration register ADC_CFG1 is set to 1, ADC is set to intermittent mode; ADC conversion can be enabled by software or by hardware-triggered event. In this mode, only one channel of one sequence is converted at a time. If DISCEN bit is cleared, all channels of one sequence will be converted at a time.

For example:

- DISCEN bit is set to 1, and the channel sequence is 0, 1, 5
  - 1st trigger, Channel 0 is converted and generates an EOCFLG event
  - 2nd trigger, Channel 1 is converted and generates an EOCFLG event
  - 3rd trigger, Channel 5 is converted and generates an EOCFLG event
- DISCEN bit is set to 0, and the channel sequence is 0, 1, 5
  - 1st trigger, channels 0, 1 and 5 are converted in sequence. After the conversion of each channel, an EOCFLG event will be generated. After the conversion of the whole sequence is completed, an EOSEQFLG event will be generated

21.3.4 ADC channel classification

21.3.4.1 Analog input channel introduced by GPIO pin

In total, 16 channels are connected to ADC_IN0…ADC_IN15.

21.3.4.2 Internal analog 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
- Enable by TSEN bit of configuration register ADC_CCFG
- Select sampling time

Internal reference voltage VREFINT

- The internal reference voltage is used to provide a stable voltage output for ADC
- Internal reference voltage V_{REFINT} selects ADC1_IN17 input channel

Internal power monitoring

The internal connection to the ADC_IN18 channel allows reading voltage states via ADC_ANA_SWITCH configuration, including VDD_FLASH_1.7_32K, VDD_FLASH_1.7_256K, and VDD_CORE_1.2.

21.3.5 External Trigger and Trigger Polarity

The external trigger event can be selected by configuring EXTTRGSEL bit of register ADC_CFG1.

Table 81 External Trigger

Trigger source	EXTTRGSEL	Trigger type
TMR1_TRGO	000	Internal signal generated by on-chip timer
TMR1_CC4	001	
TMR2_TRGO	010	
TMR3_TRGO	011	
TMR4_TRGO	100	
Reserved	101	
Reserved	110	
Reserved	111	-

When the bit EXTPOLSEL≠"0b00" for the register ADC_CFG1, the external event can trigger conversion on its selected polarity.

Table 82 Trigger Polarity Configuration

EXTPOLSEL	Source
00	Disable trigger detection
01	Detection on rising edge
10	Detection on falling edge
11	Detection on both rising edge and falling edge

21.3.6 Data register

The data can be left-aligned or right-aligned, which is determined by DALIGCFG bit of configuration register ADC_CFG1; when DALIGCFG is set to 0, it means right-aligned, and if DALIGCFG is set to 1, it means left-aligned. ADC conversion results can be left-aligned or right-aligned and stored in 16-bit data register.

21.3.7 Programmable conversion frequency division

Reducing the frequency division can improve the conversion time and 12, 10, 8 or 6-bit modes can be selected by configuring DATARESCFG[1:0] bit of register ADC_CFG1.

Table 83 Conversion Time of t_{SAR} Related to Conversion Resolution

DATARESCFG bit	t_{SAR}	t_{SAR}(ns)@f_{ADC} =14 MHz	$t_{SMPL(min)}$	t_{ADC}	t_{ADC}(ns)@f_{ADC}=14MHz
6	7.5	535 ns	1.5	9	643 ns
8	9.5	678 ns	1.5	11	785 ns
10	11.5	821 ns	1.5	13	928 ns
12	12.5	893 ns	1.5	14	1000 ns

21.3.8 Interrupt

Table 84 ADC Interrupt

Interrupt event	Event flag	Enable control
End of conversion	EOCFLG	EOCIEN
End of sequence conversion	EOSEQFLG	EOSEQIEN
End of sampling phase	EOSMPFLG	EOSMPIEN
ADC ready	ADCRDYFLG	ADCRDYIEN
Overrun	OVREFLG	OVRIEN
Analog watchdog state reset	AWDFLG	AWDIEN

21.3.9 ADC overrun

ADC overrun means when the converted data is not read by DMA or CPU on time, another converted data will take effect.

When EOCFLG bit is 1 but another new conversion has been completed, an overrun event will occur, and OVREFLG bit of register ADC_STS will be set to 1; if OVRIEN bit is set to 1, an overrun interrupt will be generated.

It is determined by OVRMAG bit of configuration register ADC_CFG1 that the data in the ADC data register are held or overwritten when an overrun event occurs:

- OVRMAG is 0: When an overrun event is detected, old data will be held in ADC_DATA register
- OVRMAG is set to 1: When an overrun event is detected, ADC_DATA register will overwrite the data by the last converted data

Note: When using DMA in continuous conversion mode, disabling the DMA channel during operation will cause the ADC to set the OVREFLG flag. Re-enabling the DMA channel at this point may still allow DMA transfers to continue.

21.3.9.1 Overrun Events with DMA Involvement

In ADC continuous conversion mode, when a DMA request is generated for ADC data transfer, a DMA transfer error or closing the DMA channel will cause an ADC overrun event. The OVREFLG bit in the ADC_STS register is set to 1. In this case, the ADC will continue to generate DMA requests. The ADC should be reinitialized; otherwise, the transmitted data may be corrupted.

21.3.10 Data conversion management

21.3.10.1 Data conversion management without DMA involvement

The software controls data conversion. Every time the conversion is completed, EOCFLG bit will be set to 1, and the conversion results can be read from ADC_DATA register. OVRMAG bit in ADC_CFG1 register should be 0.

21.3.10.2 Data conversion management without DMA involvement and overrun

When one or more channels are converted and each conversion result does not need to be read, OVRMAG bit will be set to 1, the overrun event cannot prevent ADC conversion and the register ADC_DATA only saves the last converted data.

21.3.10.3 Data conversion management by DMA

Transmission by DMA can be used to transmit the conversion results from the data register to the memory in time to prevent loss of the conversion results in the ADC_DATA register.

DMA can be enabled by setting DMAEN bit of the register ADC_CFG1 to 1. After each conversion, a DMA request will be generated to transmit the converted data of data register to the memory.

Due to the DMA failing to respond to the DMA transfer request in a timely manner, an overflow occurs (it means OVR=1). The ADC will stop generating DMA requests, and the data corresponding to new conversions will not be

transmitted via DMA. This means that all data already transferred to the RAM can be considered valid.

DMA mode is selected by DMACFG bit of configuration register ADC_CFG1:

- When DMACFG is 0, DMA is in single mode
  - DMA programming is used to transmit the fixed-length data
  - In this mode, ADC will generate a DMA request every time it converts data effectively. When ADC conversion is restarted, ADC will stop generating DMA request
- When DMACFG is set to 1, DMA is in circular mode
  - DMA programming is in circular mode or double-buffer mode
  - In this mode, when ADC conversion is started again and the converted data is valid, a DMA request will be generated

21.3.11 Low-power characteristics

21.3.11.1 Automatic delay conversion mode

This mode is used to simplify the software and optimize the application program performance during low-speed running, and ADC overrun may occur easily.

Set WAITCEN of configuration register ADC_CFG1 to 1, enable the automatic delay conversion mode, and new ADC conversion will start only after the data in ADC data register are read, which is a method of adaptive ADC speed and adaptive system reading ADC data speed.

21.3.11.2 Automatic shutdown mode

This mode can greatly reduce the power consumption of application, and is suitable for applications with relatively few conversions or long conversion request time interval. Automatic shutdown mode can be used in combination with automatic delay conversion mode in low-frequency application.

Automatic shutdown mode can be enabled by setting AOEN bit of configuration register ADC_CFG1 to 1. When AOEN bit is set to 1 and there is no ADC conversion, it will be powered off automatically, and when the conversion is started, ADC will be woken up automatically.

Note: In Auto-off mode, re-enabling the ADC will not set the ADC READY flag.

21.4 Register Address Mapping

Table 85 ADC Register Address Mapping

Register name	Description	Offset address
ADC_STS	ADC status register	0x00
ADC_IEN	ADC interrupt enable register	0x04
ADC_CTRL	ADC control register	0x08

Register name	Description	Offset address
ADC_CFG1	ADC configuration register 1	0x0C
ADC_CFG2	ADC configuration register 2	0x10
ADC_SMPTIM	ADC sampling time register	0x14
ADC_AWDT	ADC watchdog threshold register	0x20
ADC_CHSEL	ADC channel selection register	0x28
ADC_DATA	ADC data register	0x40
ADC_ANA_SWITCH	ADC analog power monitoring register	0x300
ADC_CCFG	ADC general-purpose configuration register	0x308

21.5 Register Functional Description

21.5.1 ADC state register (ADC_STS)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ADCRDYFLG	RC_W1	ADC Ready Flag 0: ADC not ready 1: ADC has been ready to start conversion In Auto-Off mode, re-enabling the ADC does not set the ADC READY flag.
1	EOSMPFLG	RC_W1	End of Sampling Flag This bit is set to 1 by hardware and cleared by software 0: Not in the phase of end of sampling 1: Reach the condition for end of sampling phase
2	EOCFLG	RC_W1	End of Conversion Flag This bit is set to 1 by hardware and cleared by software 0: Conversion does not end 1: Conversion ends
3	EOSEQFLG	RC_W1	End of Sequence Flag This bit is set to 1 by hardware and cleared by software 0: Sequence conversion not completed 1: Sequence conversion completed
4	OVREFLG	RC_W1	ADC Overrun Event Flag This bit is set to 1 by hardware and cleared by software 0: No overrun event 1: Overrun event occurred
6:5	Reserved		

Field	Name	R/W	Description
7	AWDFLG	RC_W1	Analog Watchdog Flag This bit is set to 1 by hardware and cleared by software, indicating whether an analog watchdog event occurs. 0: Not occur 1: Occurred
31:8	Reserved		

21.5.2 ADC interrupt enable register (ADC_IEN)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ADCRDYIEN	R/W	ADC Ready Interrupt Enable 0: Disable 1: Enable
1	EOSMPIEN	R/W	End of Sampling Flag Interrupt Enable 0: Disable 1: Enable
2	EOCIEN	R/W	End of Conversion Interrupt Enable 0: Disable 1: Enable
3	EOSEQIEN	R/W	End of Conversion Sequence Interrupt Enable 0: Disable 1: Enable
4	OVRIEN	R/W	Overrun Interrupt Enable 0: Disable 1: Enable
6:5	Reserved		
7	AWDIEN	R/W	Analog Watchdog Interrupt Enable 0: Disable 1: Enable
31:8	Reserved		

Note: The value of the ADC_IEN register can only be modified when STARTCEN=0.

21.5.3 ADC control register (ADC_CTRL)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	ADCEN	R/S	ADC Enable This bit is set to 1 by software and cleared by hardware. 0: Disable 1: Enable Note: ADCEN bit can be set by software only when all bits of ADC_CTRL register are 0.

Field	Name	R/W	Description
1	ADCD	R/S	ADC Disable This bit is set to 1 by software and cleared by hardware. 0: Invalid 1: Disable ADC, and enter power-down mode Note: ADCD bit can be set by software only when ADCEN=1 and STARTCEN=0.
2	STARTCEN	R/S	ADC Start Conversion Enable This bit is set to 1 by software and cleared by hardware. 0: Disable ADC conversion 1: Enable ADC conversion Note: STARTCEN bit can be set by software only when ADCEN=1 and ADCD=0.
3	Reserved		
4	STOPCEN	R/S	ADC Stop Conversion Enable This bit is set to 1 by software and cleared by hardware. 0: Invalid 1: Stop ADC conversion Note: This bit can be set by software only when STARTCEN=1 and ADCD=0.
30:5	Reserved		
31	CAL	R/S	ADC Calibrate This bit is set to 1 by software and cleared by hardware. 0: Calibration is completed 1: Start calibration Note: CAL bit can be set by software only when ADC is disabled.

21.5.4 ADC configuration register 1 (ADC_CFG1)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	DMAEN	R/W	DMA Enable 0: Disable 1: Enable Note: Software is only allowed to write to this bit when STARTCEN = 0.
1	DMACFG	R/W	DMA Mode Configure This bit is valid only when DMAEN=1. 0 : DMA single mode 1 : DMA circular mode
2	SCANSEQDIR	R/W	Scan Sequence Direction Configure 0: Scan forward (from CHSEL0 to CHSEL16) 1: Scan backward (from CHSEL16 to CHSEL0)

Field	Name	R/W	Description
4:3	DATARESCFG	R/W	Data Resolution Configure 00: 12 bits 01: 10 bits 10: 8 bits 11: 6 bits
5	DALIGCFG	R/W	Data Alignment Configure 0: Right-aligned 1: Left-aligned
8:6	EXTTRGSEL	R/W	Select external trigger event (External Trigger Event Select) These bits are used to select the external event for triggering ADC conversion. 000: Event 0 001: Event 1 010: Event 2 011: Event 3 100: Event 4 101: Event 5 110: Event 6 111: Event 7
9	Reserved		
11:10	EXTPOLSEL	R/W	External Trigger Enable and Polarity Select 00: Hardware trigger detection is closed (conversion can be started by software) 01: Hardware trigger detection on rising edge 10: Hardware trigger detection on falling edge 11: Hardware trigger detection on both rising and falling edges Avoid modifying the ADC trigger edge after the ADC is enabled, as this may lead to glitches or unexpected behaviors.
12	OVRMAG	R/W	Overrun Management Mode 0: When an overrun event is detected, ADC_DATA register saves previous data 1: When an overrun event is detected, ADC_DATA register saves the last converted data
13	CMODESEL	R/W	Select Single/Continuous Conversion Mode 0 : Single conversion mode 1 : Continuous conversion mode
14	WAITCEN	R/W	Wait Conversion Mode Enable 0 : Disable 1 : Enable
15	AOEN	R/W	Auto-off Mode Enable 0: Disable 1: Enable

Field	Name	R/W	Description
16	DISCEN	R/W	Discontinuous Mode Enable 0 : Disable 1 : Enable
21:17	Reserved		
22	AWDCH EN	R/W	Enable The Watchdog on A Single Channel or on All Channels 0: Enable analog watchdog on all channels 1: Enable analog watchdog on a single channel
23	AWDEN	R/W	Analog Watchdog Enable 0: Disable 1: Enable
25:24	Reserved		
30:26	AWDCH SEL	R/W	Analog Watchdog Channel Select These bits are used to configure the input channel for the analog watchdog to monitor ADC 00000: Channel 0 00001: Channel 1 ….. 10010: Channel 18 Other values: Reserved, not used Note: The channel selected by AWDCHSEL bit must be written to CHSEL register
31	Reserved		

Note: These bits can be rewritten only when STARTCEN=0 (confirming no ongoing conversion).

21.5.5 ADC configuration register 2 (ADC_CFG2)

Offset address: 0x10

Reset value: 0x0000 1000

Field	Name	R/W	Description
29:0	Reserved		
31:30	CLKCFG	R/W	ADC Clock Mode Configure 00: ADCCLK (asynchronous clock mode) 01: PCLK/2 (synchronous clock mode) 10: PCLK/4 (synchronous clock mode) 11: Reserved Note: The software allows writing these bits only when ADC is disabled.

21.5.6 ADC sampling time register (ADC_SMPTIM)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
2:0	SMPCYCSEL	R/W	Sampling Cycles Select 000: 1.5 ADC clock cycles 001: 7.5 ADC clock cycles 010: 13.5 ADC clock cycles 011: 28.5 ADC clock cycles 100: 41.5 ADC clock cycles 101: 55.5 ADC clock cycles 110: 71.5 ADC clock cycles 111: 239.5 ADC clock cycles Note: These bits can be rewritten only when STARTCEN=0.
31:3	Reserved		

21.5.7 ADC watchdog threshold register (ADC_AWDT)

Offset address: 0x20

Reset value: 0x0FFF 0000

Field	Name	R/W	Description
11:0	AWDLT[11:0]	R/W	Analog Watchdog Low Threshold
15:12	Reserved		
27:16	AWDHT[11:0]	R/W	Analog Watchdog High Threshold
31:28	Reserved		

Note: These bits can be rewritten only when STARTCEN=0.

21.5.8 ADC channel selection register (ADC_CHSEL)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
17:0	CHxSEL	R/W	Channel-x Select 0: Input channel x is not selected as conversion channel 1: Input channel x is selected as conversion channel
31:18	Reserved		

Note: These bits can be rewritten only when STARTCEN=0.

21.5.9 ADC data register (ADC_DATA)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	CDATA[15:0]	R	Converted Data These bits are read-only. Include the conversion result values of last conversion channel. CDATA[6:0] value is calibration factor only when calibration is completed.
31:16	Reserved		

21.5.10 ADC Analog Power Detection Register (ADC_ANA_SWITCH)

Offset address: 0x300

Reset value: 0x0000 0000

Field	Name	R/W	Description
1:0	ADC_ANA_SWITCH	R/W	ADC_IN18 Analog Power Detection select 00: 1.7 V Flash regulator output (FLASH_SMALL) 01: 1.7 V Flash regulator output (FLASH_LARGE) 10: 1.2 V core regulator output (VDD_LV) 11: Reserved
31:2	Reserved		

Note: This bit can be rewritten only when STARTCEN=0.

21.5.11 ADC general-purpose configuration register (ADC_CCFG)

Offset address: 0x308

Reset value: 0x0000 0000

Field	Name	R/W	Description
21:0	Reserved		
22	VREFEN	R/W	V_{REFINT} Enable 0: Disable 1: Enable
23	TSEN	R/W	Temperature Sensor Enable 0: Disable 1: Enable
31:24	Reserved		

Note: This bit can be rewritten only when STARTCEN=0.

22 True Random Number Generator (TRNG)

22.1 Introduction

TRNG is a random number generator which provides a 32-bit random number in the master reading based on continuous analog noise.

22.2 Main Characteristics

(1) Provides 32-bit random numbers generated by the analog generator

(2) The interval between two consecutive random numbers is 40 clock signal cycles

(3) Monitors TRNG entropy to mark abnormal behaviors

22.3 Functional Description

The true random number generator is implemented by analog circuit. This circuit provides seeds for the linear feedback shift register to generate 32-bit random numbers.

Multiple ring oscillators form an analog circuit, and the seeds are generated by XOR operation through the frequency output by the oscillator. TRNG HCLK is a dedicated clock of TRNG_LFSR, and it provides clock information for it, so the quality of random numbers is related to the frequency of HCLK. After TRNG_LFSR introduces a large number of seeds, the content will be transferred to TRNG_DATA register. The system will monitor the seeds. The status bit in TRNG_STS register indicates the abnormal sequence occurring on the seeds. An interrupt will be generated when an error is detected.

22.3.1 Enable TRNG

The setting sequence of enabling TRNG is as follows:

(1) Enable interrupt and an interrupt will be generated when the random number is ready or an error occurs.

(2) A random number will be generated when TRNG_CTRL[TRNGEN]=1. At this time, the analog part, TRNG_LFSR and error detector will be activated.

(3) At the time of each interrupt, when CLKERINT bit and FSINT bit of TRNG_STS register are set to 0 and DATARDY=1, TRNG_DATA register can be read.

The first random number generated after TRNGEN bit is set shall not be used, and it shall be saved for comparison with the next random number. Each

random number needs to be compared with the previous random number. If any pair is equal, it means that the continuous random number generator test fails.

22.3.2 Error state

22.3.2.1 Seed error

In case of a seed error, the interrupt random number will be generated as long as FSCSTS=1. Since the entropy may be insufficient, if there are already data in TRNG_DATA register, the generated interrupt random number cannot be used.

TRNGEN bit shall be set after FSINT bit is cleared to reinitialize and restart TRNG.

22.4 Register Address Mapping

Table 86 TRNG Register Address Mapping

Register name	Description	Offset address
TRNG_CTRL	TRNG control register	0x00
TRNG_STS	TRNG status register	0x04
TRNG_DATA	TRNG data register	0x08

22.5 Register Functional Description

22.5.1 TRNG Control Register (TRNG_CTRL)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
1:0	Reserved		
2	TRNGEN	R/W	TRNG Enable 0: Disable 1: Enable
3	INTEN	R/W	Interrupt Enable 0: Disable 1: Enable; when any of DATARDY bit, CLKERINT bit and FSINT bit in TRNG_STS register is set to 1, an interrupt will be pending
31:4	Reserved		

22.5.2 TRNG Status Register (TRNG_STS)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	DATARDY	R	Data Ready 0: TRNG_DATA register is not ready, and the random data is not available 1: TRNG_ DATA register is ready, and the random data is available An interrupt will be pending when INTEN=1. After reading TRNG_DATA register, this bit will be cleared to zero until a new valid value is figured out.
1	Reserved		
2	FSCSTS	R	Faulty Sequence Current Status 0: Sequence error is not detected. If FSINT bit is set to 1, it means that a sequence error is detected and has recovered to normal. 1: More than 64 0/1 or more than 32 alternate 0 and 1 are detected
5:3	Reserved		
6	FSINT	RC_W0	Faulty Sequence Interrupt Status 0: Faulty sequence is not detected 1: More than 64 0/1 or more than 32 alternate 0 and 1 are detected This bit can be set at the same time with FSCSTS bit and be cleared by writing 0. An interrupt will be pending when INTEN=1.
31:7	Reserved		

22.5.3 TRNG Data Register (TRNG_DATA)

Offset address: 0x08

Reset value: 0x0000 0000

This register is read-only and provides 32-bit random number when reading.

This register can be read only when DATARDY bit is set to 1; after reading, this register will provide new random number within 40 PLLCLK clock cycles.

Field	Name	R/W	Description
31:0	DATA	R	Random Data 32-bit random data

23 Encryption Module (AES256)

23.1 Introduction

AES256 implements the AES256 (Advanced Encryption Standard) encoding algorithm, also known as the Rijndael algorithm.

AES256 supports three keys in different lengths:

- AES-128
- AES-192
- AES-256

AES256 supports three modes:

- ECB (Electronic Codebook)
- CBC (Cipher Block Chaining)
- CTR (Counter)

AES256 supports two algorithms:

- CMAC algorithm
- MP-compress algorithm

The data lengths for the input block, output block, and state are all 128 bits. This is represented by Nb = 4, reflecting the number of 32-bit words in the state.

The length of the cipher key, K, can be 128, 192, or 256 bits. The key length is represented by Nk = 4, 6, 8, reflecting the number of 32-bit words in the cipher key.

The number of rounds required during the algorithm's execution depends on the key length. The number of rounds is represented by Nr, where: When Nk = 4, Nr = 10; when Nk = 6, Nr = 12; when Nk = 8, Nr = 14.

23.2 Main Characteristics

(1) Supports three keys in different lengths, including128-bit, 192-bit, and 256-bit.

(2) Supports encryption, decryption, and key expansion.

(3) Supports ECB mode, CBC mode, and CTR mode.

In ECB mode, each data block is encrypted/decrypted using the same key.

In CBC encryption mode, the input plaintext is obtained by XOR-ing the current plaintext with the previous ciphertext. In CBC decryption mode, the current decrypted output plaintext is obtained by XOR-ing the previous ciphertext with the decrypted result. If the user selects CBC

mode, the initial value must be written to the IV register before each operation according to the requirements above.

(4) Supports the CMAC algorithm, including subkey generation and MAC generation.

(5) Supports the MP-compress algorithm.

23.3 Programming Guide

23.3.1 ECB Mode

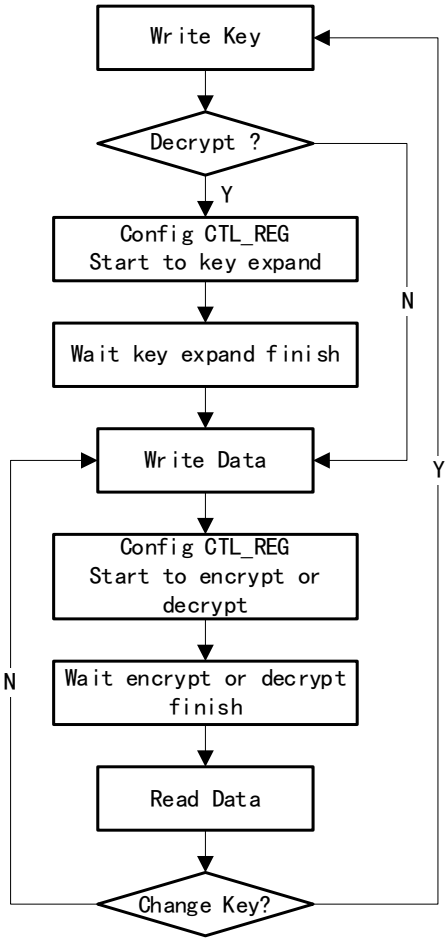
The recommended software configuration flow is as follows:

(1) Write the data and key to the corresponding registers.

(2) CTRL Configuration:
- Encryption: START + Encrypt
- Decryption: START + Key Expansion -> START + Decrypt

(3) Wait for the computation to complete, then read out the result.

Figure 87 Software Configuration Flow for ECB Mode



23.3.2 CBC Mode

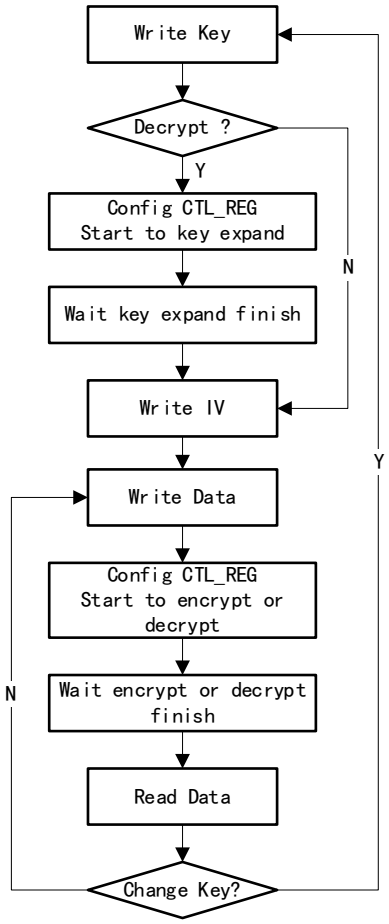
The recommended software configuration flow is as follows:

(1) Write the data, key and IV to the corresponding registers.

(2) CTRL Configuration:
- Encryption: START + Encrypt
- Decryption: START + Key Expansion -> START + Decrypt

(3) Wait for the computation to complete, then read out the result.

Figure 88 Software Configuration Flow for CBC Mode



23.3.3 CTR mode

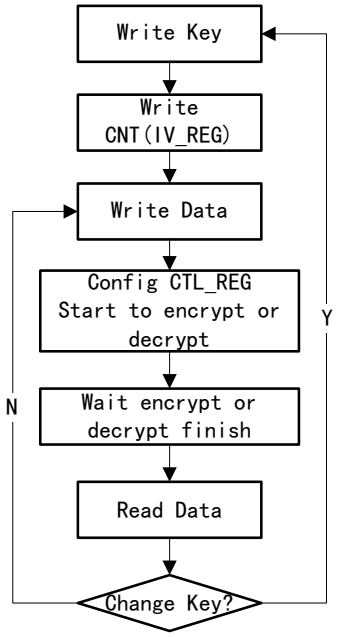
The recommended software configuration flow is as follows:

(1) Write the data, key and counter (match with IV_REG) to the corresponding registers.

(2) CTRL Configuration:
- Encryption: START + Encrypt
- Decryption: START + Decrypt

(3) Wait for the computation to complete, then read out the result.

Figure 89 Software Configuration Flow for CTR Mode



23.3.4 CMAC Mode

The recommended software configuration procedures and flow chart are as follows:

(1) Reset AES256 to clear K1/K2 data.

(2) Write the data, key and IV to the corresponding registers.

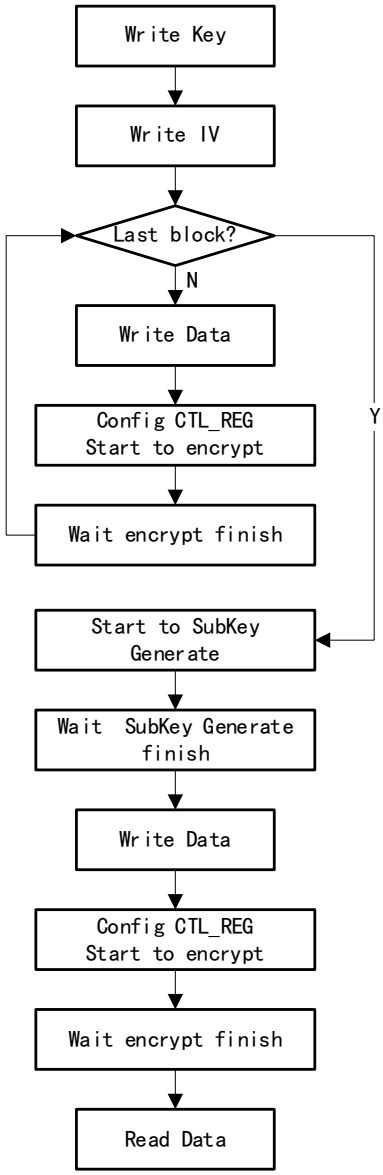
(3) Configure the CTRL register, let n = [Mlen/128], for i = 1 to n:

- If i < n, configure START + CMAC + Encrypt
- If i = n, first configure START + ECB + Encrypt + SubKG + Valid_length to generate the Subkey. Then, configure START + CMAC + Encrypt + Valid_length to generate the final MAC.

Note: Mlen = the bit length of the input string. For example, for the input string 89abcdef, Mlen = 32.

(4) Wait for the computation to complete, then read out the result.

Figure 90 Software Configuration Flow for CMAC Mode



23.3.5 MPC Mode

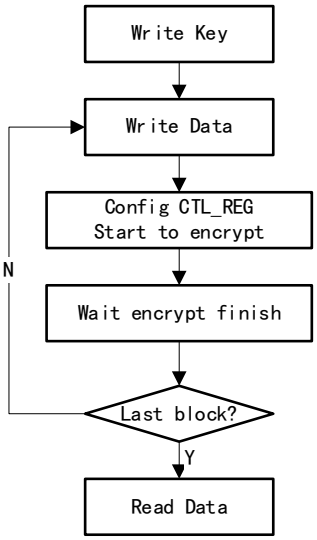
The recommended software configuration procedures and flow chart are as follows:

(1) Write the data and key (All 256 bits are “0”) to the corresponding registers.

(2) Configure CTRL, set START + encrypt +MPC.

(3) Wait for the computation to complete, then read out the result.

Figure 91 Software Configuration Flow for MPC Mode



23.4 Register Address Mapping

Table 87 AES256 Register Address Mapping

Register name	Description	Offset address
Datain_0-3	Data input register: 0-31	0x00
Datain_4-7	Data input register: 32-63	0x04
Datain_8-B	Data input register: 64-95	0x08
Datain_C-F	Data input register: 96-127	0x0C
Key_0	Key register: 0-31	0x10
Key_1	Key register: 32-63	0x14
Key_2	Key register: 64-95	0x18
Key_3	Key register: 96-127	0x1C
Key_4	Key register: 128-159	0x20
Key_5	Key register: 160-191	0x24
Key_6	Key register: 192-223	0x28
Key_7	Key register: 224-255	0x2C
IV_0	Initialization vector register: 0-31	0x30
IV_1	Initialization vector register: 32-63	0x34
IV_2	Initialization vector register: 64-95	0x38
IV_3	Initialization vector register: 96-127	0x3C
CTRL	AES256 control register	0x40
STATE	AES256 status register	0x44
Dataout_0-3	Data output register: 0-31	0x48

Register name	Description	Offset address
Dataout_4-7	Data output register: 32-63	0x4C
Dataout_8-B	Data output register: 64-95	0x50
Dataout_C-F	Data output register: 96-127	0x54

23.5 Register Functional Description

23.5.1 AES256 Data Input Register (Datain_0-3)

Offset address: 0x00

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	IN_0	R/W	Input data 0-31

23.5.2 AES256 Data Input Register (Datain_4-7)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	IN_1	R/W	Input data 32-63

23.5.3 AES256 Data Input Register (Datain_8-B)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	IN_2	R/W	Input data 64-95

23.5.4 AES256 Data Input Register (Datain_C-F)

Offset address: 0x0C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	IN_3	R/W	Input data 96-127

23.5.5 AES256 Key Register (KEY_0…7)

Offset address: 0x10-0x2C

Reset value: 0x0000 0000

KEY0 (Offset address: 0x10)

Field	Name	R/W	Description
31:0	KEY_0	R/W	Initial Keys 0-31

KEY1 (Offset address: 0x14)

Field	Name	R/W	Description
31:0	KEY_1	R/W	Initial Keys 32-63

KEY2 (Offset address: 0x18)

Field	Name	R/W	Description
31:0	KEY_2	R/W	Initial Keys 64-95

KEY3 (Offset address: 0x1C)

Field	Name	R/W	Description
31:0	KEY_3	R/W	Initial Keys 96-127

KEY4 (Offset address: 0x20)

Field	Name	R/W	Description
31:0	KEY_4	R/W	Initial Keys 128-159

KEY5 (Offset address: 0x24)

Field	Name	R/W	Description
31:0	KEY_5	R/W	Initial Keys 160-191

KEY6 (Offset address: 0x28)

Field	Name	R/W	Description
31:0	KEY_6	R/W	Initial Keys 192-223

KEY7 (Offset address: 0x2C)

Field	Name	R/W	Description
31:0	KEY_7	R/W	Initial Keys 224-255

23.5.6 AES256 Initialization Vector Register (IV_0…3)

Offset address: 0x30-0x3C

Reset value: 0x0000 0000

IV_0 (Offset address: 0x30)

Field	Name	R/W	Description
31:0	IV_0	R/W	Initialization Vectors 0-31

IV_1 (Offset address: 0x34)

Field	Name	R/W	Description
31:0	IV_1	R/W	Initialization Vectors 32-63

IV_2 (Offset address: 0x38)

Field	Name	R/W	Description
31:0	IV_2	R/W	Initialization Vectors 64-95

IV_3 (Offset address: 0x3C)

Field	Name	R/W	Description
31:0	IV_3	R/W	Initialization Vectors 96-127

23.5.7 AES256 Control Register (CTRL)

Offset address: 0x40

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	START	R/W	AES256 Enable When this bit is set, the AES256 module initiates operation; it remains active for only one clock cycle and is automatically cleared upon completion of the operation
1	KEY_INT_EN	R/W	Key Expand Finish Interrupt Enable 0: Disable 1: Enable
2	DATA_INT_EN	R/W	Encrypt or Decrypt Finish Interrupt Enable 0: Disable 1: Enable
3	BIG_ENDIAN	R/W	Big-Endian or Little-Endian Registers 0: Little-endian register 1: Big-endian register
5:4	KEY_LEN	R/W	Key Length 00: 128-bit key 01: 192-bit key 10: 256-bit key 11: Reserved
7:6	OPCODE	R/W	Operation Code 00: Encrypt 01: Decrypt 10: Key extension 11: Reserved
11:8	MODE	R/W	Mode 0000 : ECB mode 0001 : CBC mode 0010 : CMAC mode 0011 : MPC mode 0100 : CTR mode 0101~1111 : Reserved
18:12	Valid_length	R/W	Valid Length Padding is required for the last block when either SubKG or CMAC is set to 1.
19	SubKG	R/W	Sub-key Generation 0: Reserved 1: Enable subkey generation. This operation should be performed before processing the final data segment in CMAC mode.
31:20	Reserved		

Note: Before initiating AES256 operation, the data, key, and IV (required only in CBC mode) must first be written to the corresponding registers. Then, configure the CTRL register and set the START bit to activate the IP core. Finally, for encryption, only the encryption mode needs to be set. For decryption, key expansion should be set first, followed by setting the decryption mode.

23.5.8 AES256 Status Register (STATE)

Offset address: 0x44

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	BUSY	R	Busy Indication Bit (BUSY) 0: IP free 1: IP busy with key expansion or encoding or decoding
1	KEY_INT_FLG	R/W	Key Expand Finish Interrupt Flag Write “0” to clear KEY_INT_FLG
2	DATA_INT_FLG	R/W	Data Finish Interrupt Flag Write “0” to clear DATA_INT_FLG
31:3	Reserved		

23.5.9 AES256 Data Output Register (Dataout_0-3)

Offset address: 0x48

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	OUT_0	R/W	Output Ciphertext 0-31

23.5.10 AES256 Data Output Register (Dataout_4-7)

Offset address: 0x4C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	OUT_1	R/W	Output Ciphertext 32-63

23.5.11 AES256 Data Output Register (Dataout_8-B)

Offset address: 0x50

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	OUT_2	R/W	Output Ciphertext 64-95

23.5.12 AES256 Data Output Register (Dataout_C-F)

Offset address: 0x54

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	OUT_3	R/W	Output Ciphertext 96-127

24 Cyclic Redundancy Check Computing Unit (CRC)

24.1 Introduction

The cyclic redundancy check (CRC) computing unit can get 8/16/32-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.

24.2 Functional Description

24.2.1 Calculation method

Use CRC-32 (Ethernet) polynomial: 0x4C1 1DB7

$$(X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1)$$

24.2.2 Calculating Time

- The calculation time is four AHB clock cycles.

24.2.3 Functional characteristics

- Process 8-bit, 16-bit and 32-bit data
- Programmable CRC initial value
- Independent 32-bit input/output register
- It can be used as a general-purpose 8-bit register for temporary storage
- Reversible option of I/O data
- The data width can be dynamically adjusted to reduce the number of times of calculating and writing
- The high and low bits of input data can be inverted in order to adapt to different data storage methods (byte, half word or word, little-endian and big-endian system)
- Word or byte calculation can be performed, depending on the different data formats written
- Have input buffer to reduce wait cycles and avoid bus blocking

CRC unit contains a 32-bit read/write register CRC_DATA, used to write new data and give CRC computing results. 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). CRC_Data can access word or right-aligned half word or right-aligned bytes, while other registers can only access 32 bits.

24.3 Register Address Mapping

Table 88 CRC Register Address Mapping

Register name	Description	Offset address
CRC_DATA	Data register	0x00
CRC_INDATA	Independent data register	0x04
CRC_CTRL	Control register	0x08
CRC_INITVAL	CRC initial value register	0x10

24.4 Register Functional Description

24.4.1 Data register (CRC_DATA)

Offset address: 0x00

Reset value: 0xFFFF FFFF

Field	Name	R/W	Description
31:0	DATA	R/W	32-bit Data As an input register: Store the new data of CRC calculator when writing. As an output register: Return the results of CRC computing when reading.

24.4.2 Independent data register (CRC_INDATA)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
7:0	INDATA	R/W	Independent 8-bit Data Can be used for temporary storage of 1-byte data. CRC rest 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 CRC calculation and can store any data.

24.4.3 Control register (CRC_CTRL)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	RST	R/S	Reset CRC Calculation Unit Set the data register to 0xFFFF FFFF. It can only set this bit, which will be automatically cleared to 0 by hardware. 0: Invalid 1: Reset Register CRC_DATA
4:1	Reserved		

Field	Name	R/W	Description
6:5	REVI	R/W	Input Data Reverse Reverse the input data in different units. 00: Not reverse 01: In byte 10: In half-word 11: In full-word
7	REVO	R/W	Output Data Reverse 0: Not reverse 1: Reverse
31:8	Reserved		

24.4.4 CRC Initial Value Register (CRC_INITVAL)

Offset address: 0x10

Reset value: 0xFFFF FFFF

Field	Name	R/W	Description
31:0	VALUE	R/W	Initial CRC Value The CRC initial value is programmable, and this bit is used to set the initial value of CRC.

25 Real-Time Clock (RTC)

25.1 Introduction

The RTC contains BCD-encoded sub-second, time, and date registers, along with their corresponding alarm registers. It supports clock calibration function and time compensation.

25.2 Main Characteristics

(1) Timebase unit

(2) Clock calibration

(3) Subsecond, time and date

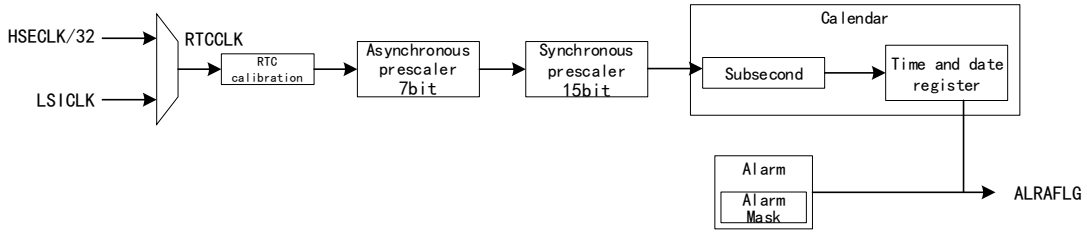
(4) Time error compensation

(5) Alarm (subsecond, time and date mask)

(6) Multiple interrupt control

25.3 Structure Block Diagram

Figure 92 RTC Structure Block Diagram



25.4 Functional Description

25.4.1 Timebase unit

Clock source

RTC has two clock sources RTC_CLK:

- Internal LSICLK
- External HSECLK crystal oscillator with 32 divided frequency

Different clock sources are configured via the RTCSRCSEL field in the RCM_RTCCTRL register of the Clock Controller (RCM).

Prescaler

The power consumption of RTC peripheral should be minimized as far as possible. Considering the power consumption, dual prescalers, 7-bit asynchronous prescaler APSC and 15-bit synchronous prescaler SPSC, are used in RTC.

RTC_CLK first passes through the asynchronous prescaler, and the clock after frequency division reaches the synchronous prescaler. Two prescalers can be reasonably configured to generate a 1Hz clock to provide date. When the prescaler is used, it is suggested that the asynchronous prescaler should be adjusted as high as possible to reduce power consumption. The synchronous prescaled value can also be used as the reload value of the subsecond counter.

25.4.2 Clock calibration

Clock synchronization

RTC can realize clock synchronization according to external high-precision clock and the register RTC_SHIFT. The deviation between RTC clock and external clock is detected mainly by acquiring the timestamps of subsecond time period twice. Since the synchronous prescaled value is used as the reload value of the subsecond counter, and the SFSEC bit of register RTC_SHIFT only works in the subsecond counter, the SFSEC bit can be adjusted to finely tune the RTC clock and increase or decrease several cycles artificially.

RTC digital calibration

RTC uses 2^{20} RTC_CLK as a calibration cycle by default. In addition, 2^{19} or 2^{18} RTC_CLK can be set as a calibration cycle through the registers AL16CFG and CAL8CFG.

When LSICLK is used as RTC_CLK clock source, the calibration cycle of RTC can be 32 s, 16 s or 8 s.

- 16 s calibration cycle: the hardware sets RECALF[0] to “0”.
- 8 s calibration cycle: the hardware sets RECALF[1:0] to “00”.

Take 32 s calibration cycle as an example, the calibration mechanism is to add or reduce some RTC_CLK signals in the calibration cycle.

- When RECALF is used, the RTC_CLK is reduced by RECALF RTC_CLK cycles every 2^{20} RTC_CLK cycles.
- When ICALFEN is used and ICALFEN=1, one RTC_CLK is added every 2^{11} RTC_CLK.
- When RECALF is used and ICALFEN is enabled, (512 * ICALFEN – RECALF) RTC_CLK are added every 2^{20} RTC_CLK.

25.4.3 RTC Write Protection

In order to prevent counting exception caused by accidental write, RTC register adopts write protection mechanism. Only when the write protection is removed,

can the register with write protection function be operated.

After power-on, RTC register will enter the write protection state by default and the protection cannot be removed by system reset. The write protection can be removed by writing special keywords ‘0xCA’ and ‘0x53’ to the register RTC_WRPROT. If other keywords are written, RTC will immediately enable write protection.

25.4.4 Calendar Register

RTC has subsecond, time and date shadow registers encoded by BCD, which are RTC_SUBSEC, RTC_TIME and RTC_DATE respectively. The current calendar can be obtained by accessing the shadow register or obtained directly from the calendar register. The time system of 24 hours and 12 hours can be selected by TIMEFCFG bit of configuration register RTC_CTRL.

RTC updates the shadow register and sets the flag bit RSFLG every two RTC_CLK cycles. When waking up from the stop or standby mode, generally the shadow register will not be updated, which requires waiting for up to two RTC_CLK cycles. The reset of shadow register is caused by system reset.

The shadow register is synchronized with f_{APB1}.

The way to read the calendar can be selected by RCMCFG bit of configuration register RTC_CTRL.

RCMCFG=0, read the calendar from the shadow register

In this mode, it recommended to set f_{APB1} >7*f_{RTC_CLK}.

After the shadow register is updated, the flag bit RSFLG will be set. The software can read the date only after the bit RSFLG is set. Every time the date is read, the RSFLG flag should be cleared manually. To ensure the normal reading of calendar value, it is required to read the shadow register twice. If the calendar obtained twice is the same, the calendar is read successfully.

When waking up from stop or standby mode, since the shadow register is not updated, the RSFLG flag should be cleared immediately.

RCMCFG=1, read the calendar from the calendar counter

When f_{APB1} is less than 7*f_{RTCCLK} or the system wakes up from low-power mode, it is recommended to read the calendar directly from the calendar counter.

If RSFLG flag bit is not set to 1 when reading the date just at the stage of change of calendar counter, it is required to read the counter twice. Therefore, it is also advised to read the calendar counter twice. When the read calendar value is the same twice, it means that the calendar is read successfully.

25.4.5 Time Compensation

Due to seasonal changes, time compensation is sometimes needed to make it more suitable for daily needs. RTC integrates time compensation unit and its summer time flag. Users can choose whether to enable time compensation according to their own needs.

By setting STCCFG bit of the register RTC_CTRL, the summer time will increase by 1 hour; by setting WTCCFG bit of the register RTC_CTRL, the winter time will decrease by 1 hour. BAKP flag is used to record whether the summer time is set.

25.4.6 Programmable Alarm

As a real-time clock, RTC integrates alarm function, and it runs mainly through alarm configuration register and alarm mask in combination with date register.

Configure the alarm and alarm mask through the registers RTC_ALRMA and RTC_ALRMASS, and the alarm mask informs RTC to pay attention to the time period of the alarm. After the alarm function is enabled, the alarm will be triggered only when the concerned time period reaches the set value. At this time, the alarm flag is set. If the alarm interrupt is enabled, the interrupt processing will be triggered.

Select “seconds” as the time period of the alarm, and only when the synchronous prescaler value is greater than 3, can the alarm operate normally.

Note: A reset caused by an RTC alarm event exiting Standby mode will not trigger the RCM reset flag.

25.4.7 RTC Reset

Certain bits in the shadow calendar registers (RTC_SUBSEC, RTC_TIME, and RTC_DATA) and the RTC status register (RTC_STS) are reset to their default values by all available system reset sources.

In contrast, the following registers are reset to their default values by an RTC domain reset and are unaffected by a system reset: RTC Current Calendar Register, RTC Control Register (RTC_CTRL), Prescaler Register (RTC_PSC), RTC Calibration Register (RTC_CAL), RTC Shift Register (RTC_SHIFT) and Alarm A Register (RTC_ALRMASS / RTC_ALRMA).

When an RTC domain reset occurs, the RTC stops operating, and all RTC registers are set to their reset values.

25.5 Register Address Mapping

Table 89 RTC Register Address Mapping

Register name	Description	Offset address
RTC_TIME	RTC Time Register	0x00

Register name	Description	Offset address
RTC_DATA	RTC Date Register	0x04
RTC_CTRL	RTC Control Register	0x08
RTC_STS	RTC Status Register	0x0C
RTC_PSC	RTC Prescaler Register	0x10
RTC_ALRMA	RTC Alarm A Register	0x1C
RTC_WRPROT	RTC Write Protection Register	0x24
RTC_SUBSEC	RTC Subsecond Register	0x28
RTC_SHIFT	RTC Shift Register	0x2C
RTC_CAL	RTC Calibration Register	0x3C
RTC_ALRMASS	RTC Alarm A Subsecond Register	0x44

25.6 Register Functional Description

25.6.1 RTC Time Register (RTC_TIME)

Offset address: 0x00

Power-on reset value: 0x0000 0000

System reset value: 0xXXXX XXXX

RTC_TIME is date time shadow register, and this register can be written only in initialization mode to be put in write protection state.

Field	Name	R/W	Description
3:0	SECU	R/W	Set the value of the second units digit, stored in BCD format
6:4	SECT	R/W	Set the value of the second tens digit, stored in BCD format
7	Reserved		
11:8	MINU	R/W	Set the value of the minute units digit, stored in BCD format
14:12	MINT	R/W	Set the value of the minute tens digit, stored in BCD format
15	Reserved		
19:16	HRU	R/W	Set the value of the hour units digit, stored in BCD format
21:20	HRT	R/W	Set the value of the hour tens digit, stored in BCD format
22	TIMEFCFG	R/W	Time Format 0: AM or 24-hour system 1: PM
31:23	Reserved		

25.6.2 RTC Date Register (RTC_DATE)

Offset address: 0x04

Reset value: 0x0000 2101

RTC_DATE is date shadow register, and this register can be written only in initialization mode to be put in write protection state.

Field	Name	R/W	Description
3:0	DAYU	R/W	Set the value of the day units digit, stored in BCD format
5:4	DAYT	R/W	Set the value of the day tens digit, stored in BCD format
7:6	Reserved		
11:8	MONU	R/W	Set the value of the month units digit, stored in BCD format
12	MONT	R/W	Set the value of the month tens digit, stored in BCD format
15:13	WEEKSEL	R/W	Week Day Units Select 000: Disable 001: Monday 010: Tuesday … 111: Sunday
19:16	YRU	R/W	Set the value of the year units digit, stored in BCD format
23:20	YRT	R/W	Set the value of the year tens digit, stored in BCD format
31:24	Reserved		

25.6.3 RTC Control Register (RTC_CTRL)

Offset address: 0x08

Power-on reset value: 0x0000 0000

System reset value: 0xXXXX XXXX

Note:

(1) The bits 7, 6 and 4 of this register can be written only in initialization mode.

(2) It is not recommended to rewrite this register when the number of hours in the date increases, which is because the correct increment of hours may be masked.

(3) The written values of STCCFG and WTCCFG will take effect from next second.

(4) This register is under write protection.

Field	Name	R/W	Description
4:0	Reserved		
5	RCMCFG	R/W	Read Calendar Value Mode Configure 0: The calendar value is read from the shadow register, and the shadow register is updated every two RTCCLK cycles 1: The calendar value is read from the calendar counter If the clock frequency of APB1 is lower than seven times of RTCCLK frequency, RCMCFG must be set to 1.
6	TIMEFCFG	R/W	Time Format Configure 0: 24-hour/day format 1: AM/PM time format
7	Reserved		

Field	Name	R/W	Description
8	ALREN	R/W	Alarm A Function Enable 0: Disable 1: Enable
11:9	Reserved		
12	ALRIEN	R/W	Alarm A Interrupt Enable 0: Disable 1: Enable
13	Reserved		
14	WUTIEN	R/W	Wakeup Timer Interrupt Enable 0: Disable 1: Enable
15	Reserved		
16	STCCFG	W	Summer Time Change Configure The bit will always be 0 in the reading process; if this bit is set not in the initialization mode, the date time will increase by 1. 0: Invalid 1: The current time increases by 1 hour to calibrate the summer time change
17	WTCCFG	W	Winter Time Change Configure The bit will always be 0 in the reading process; if this bit is set not in the initialization mode, and HRx of RCT_TIME register is 0, this bit is invalid, and if HRx is not 0, the date time will decrease by 1. 0: Invalid 1: The current time decreases by 1 hour to calibrate the winter time change
18	BAKE	R/W	Backup Value Setup This bit indicates whether the summer time has changed and is written by the user.
31:19	Reserved		

25.6.4 RTC Status Register (RTC_STS)

Offset address: 0x0C

Power-on reset value: 0x0000 0007

System reset: 0xXXXX XXXX

This register (except RTC_STS[13:8] bit) is in write protection state.

Field	Name	R/W	Description
0	ALRWFLG	R	Alarm A Write Flag When ALREN=0 for RTC_CTRL, the value of Alarm A will change and this bit will be set to “1” by hardware. This bit is cleared by hardware in initialization mode. 0: Alarm A cannot be updated 1: Alarm A can be updated
2:1	Reserved		

Field	Name	R/W	Description
3	SOPFLG	R	Shift Operation Pending Flag 0: Not occurred 1: Occurred This flag is immediately set to 1 by hardware when a shift operation is initiated by writing to the RTC_SHIFT register. It is cleared by hardware upon completion of the corresponding shift operation. Writing to the SOPFLG has no effect.
4	INITSFLG	R	Date Initialization State Flag When the “year” field in the date is not “0”, this bit will be set by hardware. 0: Not occurred 1: Occurred
5	RSFLG	RC_W0	Date and Shadow Registers Synchronization Flag Flag） When the content in the date register is copied to the shadow registers (RTC_SUBSEC, RTC_TIME and RTC_DATE), this bit is set to 1 by hardware; when shifting operation is pending (SOPFLG=1) or is in the mode that the shadow register is ignored (RCMCFG=1), this bit is cleared by hardware in initialized mode; this bit can also be cleared by software. This bit is cleared by hardware/software in initialization mode. 0: Not synchronized 1: Synchronized
6	RINITFLG	R	Date Register Initialization Flag This bit is set to “1”, RTC is in initialization state, and the time, date and prescaler registers can be updated. 0: Not initialized 1: Initialized
7	INITEN	R/W	Initialization Mode Enable 0: Free running mode 1: Initialization mode; it can be used to program RTC_TIME, RTC_DATE and RTC_PSC. The counter stops counting, and after INITEN is reset, the counter will start counting from a new value.
8	ALRAFLG	RC_W0	Alarm A Match Flag When RTC_TIME and RTC_DATE match Alarm A Register RTC_ALRMA, this flag is set by hardware. This flag can be cleared by writing 0 by software.
15:9	Reserved		
16	RCALPFLG	R	Recalibration Pending Flag When the software writes to RTC_CAL, this bit is set to 1 automatically, and the RTC_CAL register is locked. This bit will return 0 when other new calibration setting is performed.
31:17	Reserved		

25.6.5 RTC Prescaler Register (RTC_PSC)

Offset address: 0x10

Power-on reset value: 0x007F 00FF

System reset: 0xXXXX XXXX

The register can only be written in the initialization mode, and the initialization must be completed by two independent write accesses, and the register is in write protection state.

Field	Name	R/W	Description
14:0	SPSC	R/W	Synchronous Prescaler Coefficient ck_spre frequency=ck_apre frequency/(SPSC+1)
15	Reserved		
22:16	APSC	R/W	Asynchronous Prescaler Coefficient ck_apre frequency=RTCCLK frequency/(APSC+1)
31:23	Reserved		

25.6.6 RTC Alarm A Register (RTC_ALRMA)

Offset address: 0x1C

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

This register can be written only when ALRWFLG of RTC_STS is set to 1 or in initialization mode, and it is in write protection state.

Field	Name	R/W	Description
3:0	SECU	R/W	Set the value of the minute units digit, stored in BCD format
6:4	SECT	R/W	Set the value of the second tens digit, stored in BCD format
7	SECMEN	R/W	Alarm A Seconds Mask Enable 0: If the “second” matches, set Alarm A 1: Mask the effect of the “second” value on Alarm A
11:8	MINU	R/W	Set the value of the minute units digit, stored in BCD format
14:12	MINT	R/W	Set the value of the minute tens digit, stored in BCD format
15	MINMEN	R/W	Alarm A Minutes Mask Enable 0: If the “minute” matches, set Alarm A 1: Mask the effect of the “minute” value on Alarm A
19:16	HRU	R/W	Set the value of the hour units digit, stored in BCD format
21:20	HRT	R/W	Set the value of the hour tens digit, stored in BCD format
22	TIMEFCFG	R/W	Time Format Configure 0: AM or 24-hour system 1: PM
23	HRMEN	R/W	Alarm A Hour Mask Enable 0: If the “hour” matches, set Alarm A 1: Mask the effect of the “hour” value on Alarm A
27:24	DAYU	R/W	Set the value of the day units digit, stored in BCD format
29:28	DAYT	R/W	Set the value of the day tens digit, stored in BCD format
30	WEEKSEL	R/W	Date Select 0: DAYU means days 1: DAYU means the number of weeks. In this case, DAYT [1:0] is invalid.

Field	Name	R/W	Description
31	DAYEMEN	R/W	Alarm A Date Mask Enable 0: If the date/week matches, set Alarm A 1: Mask the effect of the date/week value on Alarm A

25.6.7 RTC Write Protection Register (RTC_WRPROT)

Offset address: 0x24

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	KEY	W	Write Protection Key Value Setup This byte is written by software; read this byte and it is always 0x00. Write 0xCA and 0x53 in sequence to remove the write protection of the RTC register; when writing other values, the RTC register enters the write-protected state.
31:16	Reserved		

25.6.8 RTC Subsecond Register (RTC_SUBSEC)

Offset address: 0x28

Reset value: 0x0000 0000

Field	Name	R/W	Description
15:0	SUBSEC	R	Sub Second Value Setup SUBSEC is the value of synchronous prescaler counter. It is determined by the following formula: Subsecond value=(SPSC-SUBSEC)/(SPSC+1) After one shift operation is performed, SUBSEC may be greater than SPSC. The correct time/date is one second less than RTC_TIME/RTC_DATE.
31:16	Reserved		

25.6.9 RTC Shift Register (RTC_SHIFT)

Offset address: 0x2C

Power-on reset value: 0x0000 0000

System reset value: 0xXXXX XXXX

This register is in write protection state.

Field	Name	R/W	Description
14:0	SFSEC	W	Subtract a Fraction of a Second Setup This bit field is write-only; reading this byte always returns 0. Writes to this bit are ignored when an operation is in progress. The set SFSEC value will be added to the synchronous prescaler counter. If the counter counts down, the clock will be delayed, and the delay time is determined by the following formula: Delay (seconds)=SFSEC/(SPSC+1) When it takes effect at the same time with ADD1SECEN, advance the clock and a fraction of a second will be added; the specific added value is determined by the following formula: Advance(seconds)=(1-(SFSEC/(SPSC+1))) Conduct write operation to this bit and RSFLG bit can be cleared. The software keeps running until RSFLG is set to 1 to ensure that the value of the shadow register is synchronized with the shift time.

Field	Name	R/W	Description
30:15	Reserved		
31	ADD1SECEN	W	Add One Second Enable 0: Not add 1: The clock/date increases by one second This bit field is write-only; reading this byte always returns 0. Writes to this bit are ignored when an operation is in progress. When it takes effect at the same time with SFSEC, it can increase the value of the clock by a fraction of a second.

25.6.10 RTC Calibration Register (RTC_CAL)

Offset address: 0x3C

Power-on reset value: 0x0000 0000

System reset value: 0xXXXX XXXX

This register is in write protection state.

Field	Name	R/W	Description
8:0	RECALF	R/W	Reduced Calibration Frequency Reduce date frequency: Mask RECALF pulses within 2^{20} RTCCLK pulses (32sec if the output frequency is 32768 Hz) and the date frequency will be reduced (the resolution is 0.9537 ppm). Increase date frequency: It takes effect at the same time with ICALFEN.
12:9	Reserved		
13	CAL16CFG	R/W	16 Second Calibration Cycle Period Configure When CAL16CFG is set to 1, 16-second calibration cycle is used, and it cannot be set to 1 at the same time with CAL8CFG bit. When CAL16CFG=1, RECALF[0] is always 0.
14	CAL8CFG	R/W	8 Second Calibration Cycle Period Configure When CAL8CFG is set to 1, 8-second calibration cycle is used, and it cannot be set to 1 at the same time with CAL16CFG bit. When CAL8CFG=1, RECALF[1:0] is always 00.
15	ICALFEN	R/W	Increase Calibration Frequency Enable 0: RTCCLK pulse is not increased 1: One RTCCLK pulse is increased (the frequency increases by 488.5 ppm) every 2^{11} pulses It takes effect at the same time with RECALF, and when the resolution is high, the date frequency will be reduced. If the input frequency is 32768 Hz, the number of RTCCLK pulses added within the 32-second window is determined by the following formula: (512 * ICALFEN) – RECALF.
31:16	Reserved		

25.6.11 RTC Alarm A Subsecond Register (RTC_ALRMASS)

Offset address: 0x44

Power-on reset value: 0x0000 0000

System reset: 0xXXXX XXXX

This register can be written only when ALREN of RTC_CTRL register is reset or is in initialization mode.

This register is in write protection state.

Field	Name	R/W	Description
14:0	SUBSEC	R/W	Sub Second Value Setup The subsecond value is compared with the value in the synchronous prescaler counter to determine whether to activate the alarm A, and only the bits from 0 to MASKSEL-1 are compared.
23:15	Reserved		
27:24	MASKSEL	R/W	Mask the Most-significant Bits Starting at This Bit Select 0000: Alarm A is not compared. The alarm is set when the second unit is increases by 1. 0001: When comparing for Alarm A, SUBSEC [14:1] are ignored, and only SUBSEC [0] is used. 0010: When comparing for Alarm A, SUBSEC [14:2] are ignored, and only SUBSEC [1:0] are used. 0011: When comparing with Alarm A, SUBSEC [14:3] are ignored, and only SUBSEC [2:0] are used. … 1100: When comparing with Alarm A, SUBSEC [14:12] are ignored, and only SUBSEC [11:0] are used. 1101: When comparing with Alarm A, SUBSEC [14:13] are ignored, and only SUBSEC [12:0] are used. 1110: When comparing with Alarm A, SUBSEC [14] is ignored, and only SUBSEC [13:0] are used. 1111: When comparing Alarm A, 15 SUBSEC bits all take part in, and the alarm can be activated only when all of them match. The synchronous counter overrun bit (Bit 15) is never compared. This bit is not 0 only after shift operation.
31:28	Reserved		

26 Hash Algorithm (SHA256)

26.1 Introduction

SHA-256 (Secure Hash Algorithm 256-bit) is a cryptographic hash function and a part of the SHA-2 family. Integrating an SHA-256 module into an MCU, it provides the system with efficient and secure hash computation capabilities, and can be applied in security-related scenarios such as data integrity verification, digital signatures, and password storage. Hardware implementation of the SHA-256 algorithm significantly improves computation speed compared to software implementation while reducing the CPU load. The maximum message length is 63 * 256 bits, with each message block having a length of 512 bits.

26.2 Main Characteristics

(1) Supports little-endian mode (0x6463 6261).

(2) Supports the SHA-256 algorithm.

(3) Supports segmented transmission.

The HASH computation module processes one data block at a time. If the message length exceeds the size of one data block, the message must be divided into multiple blocks for processing. The computation can be performed through multiple transmissions of message data; however, the first several transmissions must consist of complete data blocks, and the final data block must be padded. The total message length is transmitted with the last block.

(4) Supports single-block computation only.

During multi-block computations, the result from the previous computation must be loaded into the DIGEST register for the current hash calculation.

26.3 Register Address Mapping

Table 90 SHA Register Address Mapping

Register name	Description	Offset address
REV_REG	Version Register	0x00
CTRL_REG	Control register	0x04
STATUS_REG	Status register	0x08
BLKCNT_REG	Block Number Register	0x0C

Register name	Description	Offset address
DILH_REG	DILH Register	0x10
DILL_REG	DILL Register	0x14
DIN_REG	Input Data Register	0x18
DOUT_H0_REG	Output Data Register 0	0x1C
DOUT_H1_REG	Output Data Register 1	0x20
DOUT_H2_REG	Output Data Register 2	0x24
DOUT_H3_REG	Output Data Register 3	0x28
DOUT_H4_REG	Output Data Register 4	0x2C
DOUT_H5_REG	Output Data Register 5	0x30
DOUT_H6_REG	Output Data Register 6	0x34
DOUT_H7_REG	Output Data Register 7	0x38
DOUT_H8_REG	Output Data Register 8	0x3C
DOUT_H9_REG	Output Data Register 9	0x40
DOUT_H10_REG	Output Data Register 10	0x44
DOUT_H11_REG	Output Data Register 11	0x48
DOUT_H12_REG	Output Data Register 12	0x4C
DOUT_H13_REG	Output Data Register 13	0x50
DOUT_H14_REG	Output Data Register 14	0x54
DOUT_H15_REG	Output Data Register 15	0x58
DIGEST_REG	IV Register	0x5C

26.4 Register Functional Description

26.4.1 Version Register (REV_REG)

Offset address: 0x00

Reset value: 0x0001 0000

Field	Name	R/W	Description
7:0	MIN	R	Minor revision Indicates non-RTL changes (for example, platform, document).
15:8	MID	R	Middle revision Indicates firmware-invisible RTL changes.
23:16	MAJ	R	Major revision Indicates firmware-invisible RTL changes.
31:24	Reserved		

26.4.2 Control Register (CTRL_REG)

Offset address: 0x04

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	OP_START	R/W	Start a New Operation OP_START must be set to high after all other registers have been written. OP_START will be cleared in the next cycle.
1	IE	R/W	Whether to Enable OP_DONE Interrupt 0: Disable 1: Enable
5:2	TYPE	R/W	Hash Algorithm Types 2: SHA_256 Others: Reserved
6	INI MODE	R/W	Hash Initial Mode 0 : Initialization mode 1 : Non-initialization mode
7	PAD MODE	R/W	Hash Pad Mode 0 : Pad mode 1 : Non-pad mode
31:8	Reserved		

26.4.3 Status Register (STATUS_REG)

Offset address: 0x08

Reset value: 0x0000 0000

Field	Name	R/W	Description
0	OP_DONE	R/W	Operation Done OP_DONE is read as 1 when the operation completes, and is reset at the start of a new operation. Writing 0 to OP_DONE clears both OP_DONE and INTR.
1	BUSY	R	Hash Busy BUSY is 1 when the Hash operation is in progress. As long as BUSY is 1, any register configuration or attempt to start a new operation will be ignored.
2	INTR	R	Calculation Completed Interrupt Status
31:3	Reserved		

26.4.4 Block Number Register (BLKCNT_REG)

Offset address: 0x0C

Reset value: 0x0000 0001

Field	Name	R/W	Description
5:0	BLKN	R/W	Message Blocks to be Processed The block length is 512 bits, and this IP supports only single-block computation.
31:6	Reserved		

26.4.5 DILH Register (DILH_REG)

Offset address: 0x10

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	DILH	R/W	Length of Input Message High 32 bits Used only for the final operation, in bits.

26.4.6 DILL Register (DILL_REG)

Offset address: 0x14

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	DILL	R/W	Length of Input Message Low 32 bits Used only for the final operation, in bits.

26.4.7 Input Data Register (DIN_REG)

Offset address: 0x18

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	DIN	W	Input Message Register

26.4.8 Output Data Register y (DOUT_Hy_REG)(y=0…15)

Offset address: 0x1C+4y

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	DOUT_y	R	Output Message Register y Note: SHA256 uses H0~H7 (0x1C+4z (z=0…7))

26.4.9 IV Register (DIGEST_REG)

Offset address: 0x5C

Reset value: 0x0000 0000

Field	Name	R/W	Description
31:0	IV	W	INI Mode High for Digest When the INI_MODE signal is asserted high, the written content will be loaded into the DIGEST register for use. Note: For SHA-256, the DIGEST register must be read 8 times.

27 Chip Electronic Signature

27.1 Introduction

The chip electronic signature includes Flash capacity information of main memory and data memory as well as 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.

27.2 Functional Description

Main use of 96-bit chip ID:

- Used as serial number
- As the password, when writing the flash memory, the code and password can be combined by algorithm to improve the security of the codes in Flash.
- Used for startup configuration
- The reference number provided by the identity label is unique to any MCU series. Users cannot change the unique ID under any circumstances. According to different usages, users can choose to read the identity label in byte, half word, or full word.

27.3 Register Functional Description

27.3.1 96-bit unique chip ID

Base address: 0x1FFF F440

Offset address: 0x00

Field	Name	R/W	Description
31:0	U_ID[31:0]	R	Unique Identifier bits 31:0

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 Identifier bits 63:32

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 Identifier bits 95:64

27.3.2 Flash memory capacity register

Base address: 0x1FFF F44C

Offset address: 0x00

Field	Name	R/W	Description
15:0	PF_SIZE	R	PFLASH Memory Capacity Indicate the capacity of the main memory block for the product (in KB). Example: 0x0080 = 128 KB
31:16	DF_SIZE	R	DFLASH Memory Capacity Indicate the capacity of data memory area of the product (in KB). For example: 0x0040=64 KB

27.3.3 PID register

Base address: 0x1FFF F450

Offset address: 0x00

Field	Name	R/W	Description
7:0	P_Version	R	Product Version 0xA1
15:8	P_Series1	R	Product Series 1 0x01: G32A1085 A1065 A1045
23:16	P_Series2	R	Product Series 2 0x0: G32A1065 0x01: G32A1085 0x10: G32A1045
31:24	Reserved (0x00)		

28 Revision History

Table 91 Document Revision History

Date	Version	Revision History
February 2026	1.0	• Initial release
April 2026	1.1	• Corrected the IC2 mapping for CC2SEL bits (01 and 10 configurations) in TMR register. • Corrected the rising and falling edge descriptions for CC1POL bits (01 and 11 configurations) in TMR register. • Corrected the number of pulses in the RTC clock calibration chapter and calibration register. • Corrected the falling edge description in EINT_FTEN register. • Corrected the clock descriptions for the 0 and 1 states of HSI14RDYFLG bit in RCM_INT1 register. • Removed CMU frequency threshold detection function and related registers (RCM_FHCR, RCM_FLCR).
May 2026	1.2	• Modified the minimum operating voltage from 2.7V to 2.75V. • Modified the CAN TXBTO and TXBCF registers to be read only.

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. 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 “™” 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. Any information regarding the application of the product, Geehy hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party, 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 the sales 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 losses are caused to users due to 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 COUNTRIES 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'S PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED FOR USE AS CRITICAL COMPONENTS IN MILITARY, LIFE-SUPPORT, POLLUTION CONTROL, OR HAZARDOUS SUBSTANCES MANAGEMENT SYSTEMS, NOR WHERE FAILURE COULD RESULT IN INJURY, DEATH, PROPERTY OR ENVIRONMENTAL DAMAGE.

IF THE PRODUCT IS NOT LABELED AS "AUTOMOTIVE GRADE," IT SHOULD NOT BE CONSIDERED SUITABLE FOR AUTOMOTIVE APPLICATIONS. GEEHY ASSUMES NO LIABILITY FOR THE USE BEYOND ITS SPECIFICATIONS OR GUIDELINES.

THE USER SHOULD ENSURE THAT THE APPLICATION OF THE PRODUCTS COMPLIES WITH ALL RELEVANT STANDARDS, INCLUDING BUT NOT LIMITED TO SAFETY, INFORMATION SECURITY, AND ENVIRONMENTAL REQUIREMENTS. THE USER ASSUMES FULL RESPONSIBILITY FOR THE SELECTION AND USE OF GEEHY PRODUCTS. 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 PROVIDES THE DOCUMENT AND PRODUCTS "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 AND PRODUCTS (INCLUDING BUT NOT LIMITED TO LOSSES OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY USERS OR THIRD PARTIES). THIS COVERS POTENTIAL DAMAGES TO PERSONAL SAFETY, PROPERTY, OR THE ENVIRONMENT, FOR WHICH GEEHY WILL NOT BE RESPONSIBLE.

8. Scope of Application

The information in this document replaces the information provided in all previous versions of the document.